特　集　「定理証明，推論関係の新技術」

タブロー法とモデル生成型定理証明

Tableaux and Model Generation Theorem Proving

長谷川　隆三　九州大学大学院システム情報科学研究院
Ryuzo Hasegawa　Graduate School of Information Science and Electrical Engineering, Kyushu University.
hasegawa@ar.is.kyushu-u.ac.jp

藤田　博　（同　上）
Hiroshi Fujita
fujita@is.kyushu-u.ac.jp

越村　三幸　（同　上）
Miyuki Koshimura
koshi@is.kyushu-u.ac.jp

Keywords: clause tableau, proof condensation, folding-up, constraint, minimal model, negation as failure.

1. はじめに

タブロー法は，1950年代にBethやHintikkaが発明し，SmullyanとFittingが完成させた証明の方法であり，1960年代に発明された導出法とともに，証明システムの基礎となる体系として用いられてきている．以前は，「タブロー法は導出法に比べて効率が悪い」といわれていたが，近年のタブロー法の研究の進展 [D'Agostino 99] に伴い，現在では，そのような批判は少なくなりつつある．むしろ，タブロー法のほうが（補題などの効率化手法を取り入れた場合）目標指向性がある分，導出法より有利である，ともいわれている．

本稿では，まず，（関数記号を出現を許す一階述語論理式の）連言標準形に対するタブロー法である節タブローを概説する．紙面が限られているので，通常の論理式に対するタブロー法については省略した．興味ある読者は，[丹治 99] を参照されたい．次いで，節タブローの一種である，連結タブロー [Letz 92]，超タブロー [Baumgartner 98]，モデル生成木 [Manthey 88]，の定義を示す．これらは，節タブローの定義に制限を加えることで得られる．また，タブロー法の改善技法についての節を設けた．この技法は，タブローを用いて現実的な問題を解くうえで必須のものである．後半では，著者らの研究グループが行ってきたモデル生成法に関連した幾つかの研究テーマについての成果を述べる．

以下，記号論理学で用いられる基本的な概念を仮定する．これは，大まかに形式（syntax）的なものと意味（semantics）的なものに分けられる．前者に属するものに論理式（formula）や推論規則（inference rule）などがあり，後者に属するものにはモデル（model）がある．そして，両者を結び付けるものとしては，解釈（interpretation）がある．これらについては，[Fitting 96] の前半部分や [丹治 99] などを参照されたい．

2. 節タブロー

定理の自動証明は，論理式 F が恒真（universally valid）であることを機械的に証明する技術である．通常，F が恒真であることを示す代わりに，その否定 $\neg F$ から矛盾を導くことで証明を行う．タブロー法では，$\neg F$ を根とするタブロー（tableau）木を拡張規則（expansion rule）を用いて構成していく．そして，すべての枝が閉規則（closure rule）の条件を満たせば，$\neg F$ から矛盾が導かれたことになり，F が恒真であることがわかる．

さて，本章で述べる節タブローは，$\neg F$ の代わりに $\neg F$ の連言標準形（conjunctive normal form）を根とするタブロー木を構築する．連言標準形とは，$(L_1^1 \vee \ldots \vee L_{k_1}^1) \wedge \ldots \wedge (L_1^m \vee \ldots \vee L_{km}^m)$（各 L_i^j（$1 \leq i \leq k_j$, $1 \leq j \leq m$）はリテラル）の形をした論理式である．ここで，リテラルの選言 $L_1^i \vee \ldots \vee L_{ki}^i$ を節（clause）と呼ぶ．任意の論理式に対して，それと（充足可能性が）等価な連言標準形を機械的につくることができる [Leitsch 97].

本論文では，m 個の節の連言からなる連言標準形（$L_1^1 \vee \ldots \vee L_{k_1}^1) \wedge \ldots \wedge$（$L_1^m \vee \ldots \vee L_{km}^m$）を m 個の節からなる節集合 $\{L_1^j \vee \ldots \vee L_{ki}^j \mid 1 \leq j \leq m\}$ と同一視する．

【定義 2·1】（節タブロー）　節集合 S の節タブローは，以下のようなリテラルでラベル付けされた木である：

(i) *true* とラベル付けされた一つの節点からなる木は，S の節タブローである（初期値）．

(ii) T を S の節タブロー，N を T の葉，$L_1 \vee \ldots \vee L_n$ を $C \in S$ の異体（variant）[*1] で T と変数を共有し

*1 C と C' が変数名の違いを除いて，本質的に同じ場合，C は C' の異体という．

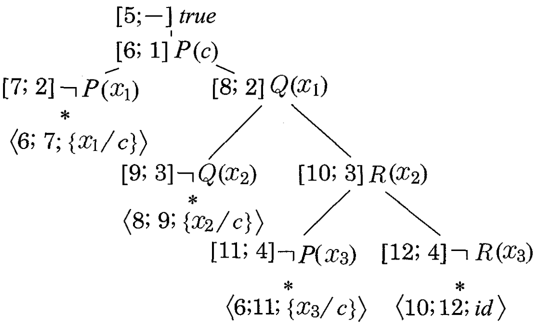

図1　$S1$のタブロー証明

ない節とする．このとき，Nを親とするn個の子節点を作り，それぞれをL_i $(i=1, ..., n)$ でラベル付けする．こうしてできた木もSの節タブローである（拡張規則）．

(iii) TをSの節タブロー，NをTの葉，M_BをTの根からNに至る枝（branch）B上に現れるリテラルの集合とする．そして，$L \in M_B$, $L' \in M_B$であるとする．このとき，Lと$\overline{L'}$ *2が最汎単一化子（most general unifier）σによって単一化可能であるとき，T中の各リテラルにσを適用して得られる木，すなわち$T\sigma$は節タブローである．このとき，Bは閉じている，という（閉規則）．

【定義2·2】（節タブロー証明）　節集合Sの節タブローTが閉じているとは，Tの根から葉に至るすべての枝が閉じていることをいう．このとき，TをSの節タブロー証明という．

[定理2·1]（節タブローの健全性と完全性）　節集合Sの節タブロー証明が存在すれば，Sは充足不能である．また逆に，Sが充足不能であれば，Sの節タブロー証明が存在する．

節タブローを構成することにより，節集合の充足可能性を判定する手法を節タブロー法と呼ぶ．

〔例2·1〕（節タブロー）　$S1$を次の四つの節からなる集合とする．

(1) $P(c)$　　(2) $\neg P(x) \vee Q(x)$

(3) $\neg Q(x) \vee R(x)$　　(4) $\neg P(x) \vee \neg R(x)$

$S1$の節タブロー$T1$を図1に示す．$T1$の各節点は，5から番号付けされている．なお1から4の番号は，$S1$の各節を参照するものとする．i番目の節点N_iに付加された$[i; j]$は，N_iが拡張規則を用いて第j節により得られたことを示している．

$T1$の根から葉に至るすべての枝を閉じることができるので，$T1$は$S1$の節タブロー証明である．各葉の下の*は，その葉に至る枝が閉じていることを表す．また，三つ組$\langle i; j; \sigma \rangle$は，その枝を，相補リテラル$i$と$j$に単一化子$\sigma$を適用して閉じたことを示している．

*2 L'が$\neg A$のとき，$\overline{L'} = A$であり，それ以外のとき，$\overline{L'} = \neg L'$と定義する．

節タブローの定義では，拡張規則に用いることのできる節は任意であるので，無節操に節を選択すると節タブローは不必要に大きくなり得る．これは，節タブローの大きな欠点の一つで，拡張規則の定義に何らかの制限を設けてこれに対処する手法が提案されている．

ここでは代表的な手法として，連結タブロー（connection tableau）と超タブロー（hyper tableau）を紹介する．

拡張規則によって得られる新しい枝のうち，

- 前者は，少なくとも一つは，
- 後者は，負リテラルでラベル付けされた葉をもつ枝のすべては，

直ちに閉じられなければならないことを要求する．

【定義2·3】（連結タブロー）　連結タブローは，定義2·1の(i)と(ii)を次のように変更することにより定義される．なお，(iii)の定義は変更せずにそのまま用いる．

(i') $L_1 \vee ... \vee L_n$を$C \in S$の任意の異体とするとき，*true*を親とするn個の節点をつくり，それぞれをL_i $(i=1, ..., n)$ でラベル付けしてできる木は，Sの連結タブローである．

(ii') TをSの連結タブロー，NをTの葉，LをNのラベル，$L_1 \vee ... \vee L_n$を$C \in S$の異体でTと変数を共有しない節とし，$\overline{L}$とあるL_i $(i \leq 1 \leq n)$ が最汎単一化子σによって単一化可能であるとする．このとき，Nを親とするn個の子節点をつくり，それぞれをL_j $(i=1, ..., n)$ でラベル付けする．こうしてできた木をT'とすると，$T'\sigma$もSの連結タブローである．

なお，新たにできるn個の枝のうち，i番目の枝は閉じていることに注意されたい．

超タブローの定義は，節の含意形式（implication form）を用いて行う．節の含意形式とは，節のリテラルの並びを，負リテラルが前方に正リテラルが後方になるように並び替えた$\neg A_1 \vee ... \vee \neg A_n \vee B_1 \vee ... \vee B_m$を$A_1 \wedge ... \wedge A_n \rightarrow B_1 \vee ... \vee B_m$と表記したものである．

【定義2·4】（超タブロー）　超タブローは，定義2·1の(ii)と(iii)を合わせて次のように変更することにより定義される．なお，(i)の定義は変更せずにそのまま用いる．

(ii'') TをSの超タブロー，NをTの葉，M_BをTの根からNに至る枝B上に現れるリテラルの集合，$A_1 \wedge ... \wedge A_n \rightarrow B_1 \vee ... \vee B_m$を$C \in S$の異体で$T$と変数を共有しない節とし，$\exists\sigma$（最汎単一化子）$\forall i (1 \leq i \leq n)(C_i \in M_B \wedge A_i\sigma = C_i\sigma)$が成り立っているものとする．このとき，$N$を親とする$n+m$個の子節点をつくり，$n$個の節点には$\neg A_i$ $(1 \leq i \leq n)$ で，m個の節点にはB_j $(1 \leq j \leq m)$ でラベル付けした木をT'とすると$T'\sigma$はSの超タブローである．

なお，新たにできる$n+m$個の枝のうちn個の枝，すな

わち，$\neg A_i$の枝はすべて閉じることに注意されたい．特に$m=0$のとき，新たにできるすべての枝は閉じることになる．

定義からわかるように，図1の節タブローは，連結タブローでもあり，超タブローでもある．

次章以降で述べるモデル生成法は，超タブローにさらに制限を加えたモデル生成木を構成する手法である．モデル生成木の節点のラベルはすべて，基礎リテラル（ground literal: 変数を含まないリテラル）でラベル付けされる．

【定義2·5】（違反節） 節$A_1\wedge\ldots\wedge A_n\rightarrow B_1\vee\ldots\vee B_m$が，基礎代入$\sigma$のもとで基礎リテラルの集合$M$に違反（violated）しているとは，$\forall i(1\leq i\leq n)\, A_i\sigma\in M\wedge\forall j(1\leq j\leq m)\, B_j\sigma\notin M$であることをいう．

【定義2·6】（モデル生成木） モデル生成木は，定義2.4の(ii″)を次のように変更することにより定義される．なお，(i)の定義は変更せずにそのまま用いる．

(ii‴) TをSのモデル生成木，NをTの葉，M_BをTの根からNに至る枝B上に現れる（基礎）リテラルの集合とし，節$(A_1\wedge\ldots\wedge A_n\rightarrow B_1\vee\ldots\vee B_m)\in S$が，基礎代入$\sigma$のもとで$M_B$に違反しているとする．このとき，$N$を親とする$n+m$個の子節点をつくり，$n$個の節点には$\neg A_i\sigma(1\leq i\leq n)$で，$m$個の節点には$B_j\sigma(1\leq j\leq m)$でラベル付けした木$T'$は，$S$のモデル生成木である．

連結タブロー，超タブロー，モデル生成木は，節タブローでもあるので三者の健全性は明らかである．一方，三者の完全性も成り立つ．

[定理2·2]（完全性） 節集合Sが充足不能であれば，Sの連結タブロー証明，超タブロー証明，モデル生成証明が存在する．

3. タブローの改善技法

本章では，タブロー証明からむだな推論を除去する証明濃縮（proof condensation）[Oppacher 88]と畳込み（folding-up）[Letz 94]について述べる．前者は，証明に関連しない部分を削除し，後者は重複した証明を回避する．

3·1 証明濃縮

タブローの葉Nを節$L_1\vee\ldots\vee L_n$で拡張し推論を進めていき，L_iの下の部分タブロー証明として$P_i\ (i=1,\ldots,n)$が得られたものと仮定する．そして，L_iがP_iのどの枝の閉規則の適用にも用いられていないものとする．このとき，Nの下にP_iを接木しても，P_iは部分タブロー証明としての条件を満たす．つまり，$L_1\vee\ldots\vee L_n$による拡張は除去できる（図2）．

今，$L_1\vee\ldots\vee L_n$による拡張の後，タブローの左から右に深さ優先で推論が進んでいく，つまり，部分タブロー

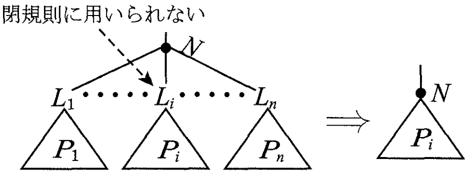

図2 証明濃縮

証明を$P_1, P_2, \ldots$の順で行っていくものとする．さて，ある$i(1\leq i\leq n)$に対して，L_iがP_iのどの葉の閉規則の適用にも用いられていないことが判明したとする．このとき，証明濃縮により，Nの下の証明として，P_iを採用することができる．つまり，$j\,(i<j\leq m)$に対する証明P_jを刈り込むことができる．証明濃縮による刈込みは容易に実装できるので，多くの証明システムに組み込まれている．

3·2 畳込み

次のような2^n個の節からなる節集合S_nを考える．

$$S_n=\{L_1\vee\ldots\vee L_n\,|\,L_i\in\{P_i,\neg P_i\}, i=1,\ldots,n\}$$

ここで，$P_i\ (i=1,\ldots,n)$は相異なる基礎アトムとする．S_nは，充足不能なのでS_nのタブロー証明が存在するが，どのようなタブロー証明でもその枝数は，$n!$個以上になることが証明されている[D' Agostino 92]．S_nは，本質的には命題論理の問題なので真理値表によって充足不能であることを示せるが，その行数は2^nである．このように問題によっては，タブロー証明は真理値表による充足不能性の判定より効率が悪い*3．

このような非効率性は，原子カット（atomic cut）規則を定義2·1に加えれば回避できる．

【定義3·1】（原子カット規則） TをSの節タブロー，NをTの葉，Pを任意のアトム（原子論理式）とする．このとき，Nを親とする2個の子節点をつくり，一方をPで，もう一方を$\neg P$でラベル付けする．こうしてできた木もSの節タブローである．

原子カットは，証明の計算量を小さくするという観点からは，非常に強力な規則であるが，アトムPは任意であるので，Pを選択する何らかの指針が実用上必要である．Letzらは，この問題に対するいくつかの解答を提案している[Letz 94]が，本節ではその中の「畳込み」という手法について触れる．これは，部分タブローから補題を生成して，その補題を利用してむだな証明を刈り込む手法，とみなすことができる．

畳込みの例を図3に示す．上図が畳込み前，下図が畳込み後のタブローである．上図は，矢印で示す葉に至る枝に閉規則を適用した後のタブローを表している．この時点で，$R(x_2)$の下の部分タブローの二つの葉に至る枝が閉じたことになる．閉じるのに用いられたリテラルの

*3 もちろん逆の場合もある．

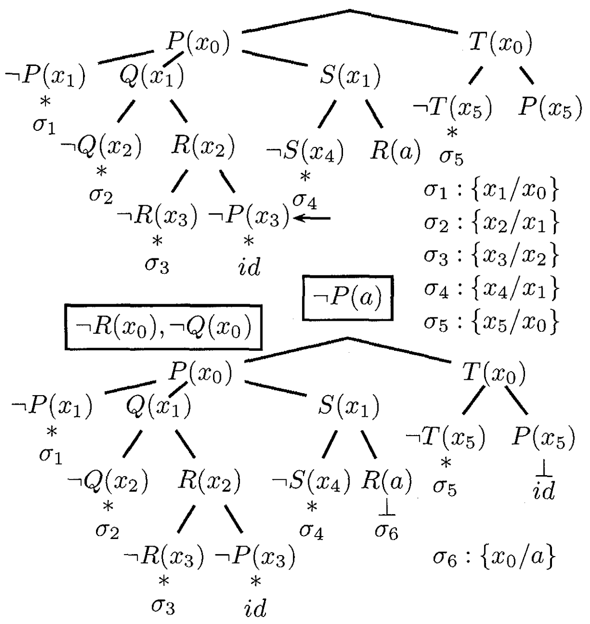

図3　畳込み前後

対は，$\langle \neg R(x_3), R(x_2) \rangle$ と $\langle \neg P(x_3), P(x_0) \rangle$ である．詳細は省くが，このことから，$S_2 \cup \{P(x_0)\} \models \neg R(x_0)$（$= \neg R(x_2)\sigma_1\sigma_2 \cdots \sigma_5$）がいえる（$S_2$ は証明しようとしている節集合）．同様に，$Q(x_1)$ の下の部分タブローも閉じていることから，$S_2 \cup \{P(x_0)\} \models \neg Q(x_0)$ がいえる．

これらは，（S_2 の証明では）$P(x_0)$ であれば $\neg R(x_0)$ および $\neg Q(x_0)$ が成り立つことを意味している．つまり，$P(x_0)$ を根にもつ部分タブローでは，$R(x_0)$ もしくは $Q(x_0)$ と単一化可能なリテラルを葉にもつような枝は直ちに閉じることができる．下図の左上の辺のラベル $\boxed{\neg R(x_0), \neg Q(x_0)}$ はこのことを示している．このように，辺または根をリテラルでラベル付けすることを畳込みと呼ぶ．これらのラベルは，節点のラベルと同じく，閉規則に用いることができる．下図の $R(a)$ に至る枝は，この畳込みによって閉規則が適用できるようになる．$\perp$ はその枝が畳込みによるリテラルによって，閉じられたことを示す．

$R(a)$ に至る枝を閉じたことにより，$P(x_0)$ の下の部分タブローの葉に至る枝がすべて閉じたことになる．上と同じ考察により，$S_2 \models \neg P(a)$（$= \neg P(x_0)\sigma_1\sigma_2 \cdots \sigma_6$）が成り立つ．こうして，根が $\boxed{\neg P(a)}$ でラベル付けされる．これにより，$P(x_5)$ に至る枝を閉じることができる．このように，畳込みを行うことにより，$R(a)$ と $P(x_5)$ の下の証明を省略できる．

モデル消去法（連結タブロー法の一種）に畳込みと類似の機能を組み込むことにより，良好な結果が得られている [Iwanuma 97]．また，モデル生成法に畳込みと証明濃縮を組み込んだシステムも作成されている [越村 00]．

4. モデル生成法の機能拡張

本章と次章では，モデル生成法に関する研究について述べる．その準備として，用語の定義とモデル生成木の性質について示す．

【定義 4·1】（完了した枝）　節集合 S のモデル生成木の枝を B，M_B を B 上に現れる基礎リテラルの集合とする．このとき，S のどの節も M_B に違反していなければ，B は完了しているという．

[定理 4·1]　節集合 S のモデル生成木の閉じていない枝を B，M_B を B 上に現れる基礎リテラルの集合とする．このとき，B が完了していれば，M_B は S のモデルである．

4·1　制 約 処 理

モデル生成法で制約充足問題を効率良く解くための拡張として，負情報を用いる手法 [白井 97]，および，区間情報を用いる手法 [Hähnle 00] を紹介する．

§1　負制約処理

1992年，モデル生成法に基づく定理証明システムMGTPは，256個の処理要素（PE）を有する並列推論マシンPIM/m上で，準群に関する未解決問題を解くことに成功した [Fujita 93]．この問題記述には，

$$p(Y, X, A) \land p(A, Y, B) \to p(B, Y, X).$$

のような節が現れるが，含意形式中に負リテラルの出現を許すものとして，対偶をとると，次のような二つの節が得られる．

$$p(Y, X, A) \land \neg p(B, Y, X) \to \neg p(A, Y, B).$$
$$p(A, Y, B) \land \neg p(B, Y, X) \to \neg p(Y, X, A).$$

もとの節にこれら対偶節をあわせて用いると，閉規則に必要となる負リテラルの導出の機会が増し，タブローの分枝数減少の可能性が高まる．これにより，通常のモデル生成法では手に負えない準群問題でも解けるようになった．

§2　区間制約処理

例えば，8人の女王問題で，$\to q(1, 1) \lor \ldots \lor q(1, 8)$ のような節を用いると，探索木の枝数は爆発的に増大する．代わりに，$\to q(1, [1, 8])$ によって上の選言と同等の内容を表現すれば，見かけ上の場合分けを減らせる可能性がある．このような形式を，ここでは**制約アトム**（*constrained atom*）と呼び，通常の**項部分**（*term part*）$q(1)$ と**制約部分**（*constraint part*）$\langle \kappa \rangle$（ここで，κ は $\{1, \ldots, 8\}$ の部分集合）の二つの部分からなると考える．

そして，モデル生成木の各節点は，制約アトム $q(c, \kappa)$ でラベル付けされる．これは，$q(c) \in \kappa$ であることを表している．推論規則としては，**モデル候補改訂**（*model candidate update*）が導入される．例えば，$q(1\ \{2, 3\})$ を含む枝を新たに $q(1\ \{1, 2\})$ で拡張したとしよう．両者は矛盾するわけではないし，互いに包摂関係にもない．

しかしながら，$q(1)$ のとり得る値の範囲が狭まり，両者の連言を表す $q(1\ \{2\})$ を生成し枝に付加するような改訂が必要となる．

4・2 ノンホーンマジックセット法

超タブロー法やモデル生成法では，正節（正リテラルのみからなる節）による拡張によって証明が始まり，負節（負リテラルのみからなる節）による拡張により証明が終了する．このような証明を上昇型証明と呼ぶ．上昇型証明には，負節の情報が証明の途中経過にまったく生かされない，という欠点がある．したがって，負節とはまったく無関連な節によるむだな拡張を行ってしまう危険性がある．

このような危険性を回避するため，ノンホーンマジックセット（NHM: Non-Horn Magic set）法が提案されている［長谷川 97］．これは，上昇型証明と下降型証明（負節から始まる証明）を融合する手法である．融合のために，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る．この変換節集合に対して，超タブロー法やモデル生成法が適用される．変換には，「幅優先」と「深さ優先」の二種があるが，ここでは幅優先のみを示す．

【定義 4・2】（幅優先 NHM 変換） 節集合 S の節 $A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_m$ を次のように二つの節に変換することを，幅優先 NHM 変換という．

$$T_B^1:\ goal(B_1) \wedge \ldots \wedge goal(B_m) \rightarrow goal(A_1) \wedge \ldots \wedge goal(A_m) \qquad (n \geq 1)$$

$$T_B^2:\ goal(B_1) \wedge \ldots \wedge goal(B_m) \wedge A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_m \qquad (n \geq 0)$$

$goal$ は S に出現しない特別な述語記号で $goal()$ の形をしたアトムを $goal$ アトムと呼び，通常のアトムと区別する．T_B^1 が下降型証明を模擬するための節，T_B^2 が負節との関連性を検査しながら上昇型証明を行う節である．

証明に関連しない節による拡張を削除するという点で，NHM 変換節の証明と証明濃縮を組み込んだ証明には類似性があるが，関連性の検査を，前者は拡張の前に，後者は拡張の後で，行うという違いがある．

4・3 極小モデル生成

節集合 S のモデル生成木の完了して閉じていない枝は，S のモデルを与える（定理 4・1）．S のすべての極小モデルは，有限であれば，モデル生成により求めることができるが，逆に，モデル生成で得られるモデルがすべて極小とは限らない．

非極小モデルを排除するためには，枝の拡張に**相補分割**（*complement splitting*）を用いる手法が有効である．これは，原子カットの機能を組み込んだ拡張規則で，拡張の際に各分枝に分岐仮定が付加される*4．これにより，非極小モデルの生成を抑制できる．しかし，すべての非極小モデルの生成を抑制するわけではない．したがって，得られたモデルが極小であるかどうかの判定が必要となる．

Bry らは，**制約下探索**（*constrained search*）でモデルの極小性の判定を行っている［Bry 96］．これは，ある順番でモデル生成木を構成していったとき，n 番目に得られるモデル M が，$n-1$ 番目までのモデルより大きくなければ，M は極小である，という性質を利用している．

Niemelä は極小モデルの性質を探求して，モデル M が**根拠検査**（*groundedness test*）に合格すれば，M が極小であることを示した［Niemelä 96］．根拠検査は，M を元に得られる節群を S に加えた節集合の充足可能性を判定することで行われる．

著者らは，相補分割に加え，分岐補題を利用することで非極小モデルの生成をさらに抑制できることを示した．分岐補題は，分岐仮定を閉規則に用いていない部分証明木より得られる．また，モデル生成木の分析により，制約下探索の負荷を軽減できる条件を示した［長谷川 01］．

*4 例えば，$A \vee B$ による拡張では，A の分枝に $\neg B$ を付加する．

5. モデル生成法の応用

5・1 失敗による否定

失敗による否定（NAF: negation as failure）は，論理プログラミングの分野で開発された最も重要な技術の一つであろう．失敗による否定を取り入れた論理プログラムは，十分強力な知識表現手段となり得る．本節では，失敗による否定を表現できる一般論理プログラム（general logic program）の安定モデル（stable model）をモデル生成法を用いて計算する技法について概説する．

一般論理プログラムとは，次のような節からなる節集合である．

$$A_1 \wedge \ldots \wedge A_m \wedge not\, A_{m+1} \wedge \ldots \wedge not\, A_n \rightarrow C$$

ここで，$A_i\ (i=1, \ldots, n)$ と C はアトムで，not が NAF を表す．$not\, A$ は，A が証明できないとき真となる．

一般論理プログラム P の各節を次のように変換し，モデル生成法でモデルを求める．

$$A_1 \wedge \ldots \wedge A_m \rightarrow (\neg \mathbf{K}A_{m+1} \wedge \ldots \wedge \neg \mathbf{K}A_n \wedge C) \vee \mathbf{K}A_{m+1} \vee \ldots \vee \mathbf{K}A_n$$

ここで，$\mathbf{K}A$ は，直感的には「A は証明される必要がある」ということを表している．こうして求めたモデル M のうち，$\forall A \in M(\neg \mathbf{K}A \notin M) \wedge \forall \mathbf{K}A \in M(A \in M)$ を満たすモデルは，P の安定モデルとなる［Inoue 92］．

5・2 法 的 推 論

法律は競合する規則を含むことが多く，これを厳密な論理システムの俎上に乗せるためには，規則間の優先順位を導入するなどの工夫が必要である．

§1　法規則の表現

法規則の多くは，拡張論理プログラミング言語によると，次のような**既定規則**（*default rule*）：

$$R{::}\ \ L_0 \Leftarrow L_1 \wedge ... \wedge L_{m-1} \wedge not\ L_{m+1} \wedge ... \wedge not\ L_n$$

を用いて表現できる．この規則において，本体のすべてのリテラル $L_i\,(0 < i \leq m)$ と $not\ L_j\,(m < j \leq n)$ が真であるとき，相反する結論を導く規則や統合性制約がなければ，頭部も真とされる．ただし，R は規則の識別子で，$L_i\,(0 \leq i \leq n)$ に現れるすべての変数を引数としてもつ．

この種の言語の意味論に関しては数多くの提案があるが，ここでは，[Kowalski 96] の手法に基づいた含意形式への変換について述べる．

§2　含意形式への変換

二つの既定規則：

$$R_1{::}\ \ L_0^1 \Leftarrow L_1^1 \wedge ... \wedge L_m^1 \wedge not\ L_{m+1}^1 \wedge ... \wedge not\ L_n^1$$

$$R_2{::}\ \ L_0^2 \Leftarrow L_1^2 \wedge ... \wedge L_k^2 \wedge not\ L_{k+1}^2 \wedge ... \wedge not\ L_q^2$$

が存在するとき，まず，R_1 を，

$$L_1^1 \wedge ... \wedge L_m^1 \wedge not\ L_{m+1}^1 \wedge ... \wedge not\ L_n^1 \wedge not\ defeated(R_1)\ \rightarrow\ L_0^1$$

のように書き換える．もとの R_1 の本体が充足され，かつ R_1 が負かされ得るという証明がない限り，頭部が導かれる，というように解釈される．述語 *defeated* は以下のような規則で定義される．

$$L_1^1\theta \wedge ... \wedge L_m^1\theta \wedge not\ L_{m+1}^1\theta \wedge ... \wedge not\ L_n^1\theta \wedge not\ defeated(R_1\theta) \wedge L_1^2\theta \wedge ... \wedge L_k^2\theta \wedge not\ L_{k+1}^2\theta \wedge ... \wedge not\ L_q^2\theta \wedge not\ R_1\theta < R_2\theta \rightarrow defeated(R_2\theta)$$

ここで，最汎単一化子 θ は，$L_0^1\theta = \neg L_0^2\theta$ を満たすものとする．また，$R_1 < R_2$ で，規則 R_2 は R_1 に優先することを表す．

NAF を含む節は，前小節のようにして K 演算子を用いた節に変換され，得られた節集合のモデル生成木を作ることにより，安定モデルを求めることができる．

ある安定モデルに現れる証明構造がすべての安定モデルに共通に現れるとき，これを**正当な言い分**（*justified argument*）と呼ぶ．さもなければ**まことしやかな言い分**（*plausible argument*）と呼ぶ．正当な言い分はいかなる論駁にも敗れることがない．一方，まことしやかな言い分のほうは競合する論駁に敗れる可能性がある．

6. おわりに

本稿では，モデル生成法に関する話題とモデル生成法を含む多くの定理証明手法の基礎となる節タブローについて解説した．タブロー法は，証明効率の点で導出法と肩を並べるほどになってきている [Hähnle 01]．加えて，導出法に比べ非標準論理を扱いやすい，という特徴がある．非標準論理は，ハードウェアやソフトウェアの検証，形式的方法，自然言語処理などと関連しており，近年のコンピュータ発展に伴いますます重要性が増している．今後は，これら応用分野との連携によって，より研究が進展していくものと予想される．

タブロー法に関する国際会議が毎年一回開催されている．最新の情報は，

http://i12www.ira.uka.de/TABLEAUX/

を参照されたい．

◇ 参 考 文 献 ◇

[Baumgartner 98] Baumgartner, P.: Hyper Tableau - The Next Generation, in Swart, de H. ed., *Theorem Proving with Analytic Tableaux and Related Methods,* Vol. 1397 of *LNAI,* pp. 60-76, Springer (1998)

[Bry 96] Bry, F. and Yahya, A.: Minimal Model Generation with Positive Unit Hyper-Resolution Tableaux, in *Theorem Proving with Analytic Tableaux and Related Methods,* Vol. 1071 of *LNAI,* pp. 143-159, Springer (1996)

[D'Agostino 92] D'Agostino, M.: Are Tableaux an Improvement on Truth-table ?, *J. of Logic, Language and Information,* Vol. 1, pp. 235-252 (1992)

[D'Agostino 99] D'Agostino, M., Gabbay, D., Hähnle, R. and Posegga, J. eds.: *HANDBOOK OF TABLEAU METHODS,* KLUWER ACADEMIC (1999)

[Fitting 96] Fitting, M.: *First-Order Logic and Automated Theorem Proving,* Springer, second edition (1996)

[Fujita 93] Fujita, M., Slaney, J. and Bennett, F.: Automatic Generation of Some Results in Finite Algebra, in *Proceedings of International Joint Conference on Artificial Intelligence* (1993)

[Hähnle 00] Hähnle, R., Hasegawa, R. and Shirai, Y.: Model Generation Theorem Proving with Finite Interval Constraints, in *Computational Logic -CL2000,* Vol. 1861 of LNAI, pp. 285-299, Springer (2000)

[Hähnle 01] Hähnle, R.: Tableaux and Related Methods, in Robinson, A. and Voronkov, A. eds., *HANDBOOK OF AUTOMATED REASONING,* Vol. 1, chapter 3, NORTHHOLLAND (2001)

[長谷川 97] 長谷川, 井上, 太田, 越村: 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, Vol. 38, No. 3, pp. 453-461 (1997)

[長谷川 01] 長谷川, 藤田, 越村: 分岐補題の抽出による極小モデル生成の効率化, 人工知能学会論文誌, Vol. 16, No. 2, pp. 234-245 (2001)

[Inoue 92] Inoue, K., Koshimura, M., and Hasegawa, R.: Embedding Negation as Failure into a Model Generation Theorem Prover, Vol. 607 of *LNAI,* pp. 400-415, Springer (1992)

[Iwanuma 97] Iwanuma, K.: Lemma Matching for a PTTP-based Top-down Theorem Prover, in McCune, W. ed., *Automated Deduction - CADE-14,* Vol. 1249 of *LNAI,* pp. 146-160, Springer (1997)

[越村 00] 越村, 長谷川: 証明の依存性解析によるモデル生成法の冗長探索の削除, 人工知能学会誌, Vol. 15, No. 6, pp. 1064-1073 (2000)

[Kowalski 96] Kowalski, R. and Toni, F.: Abstract Argumentation, *Artificial Intelligence and Law Journal,* Vol. 4, pp.275-296 (1996)

[Leitsch 97] Leitsch, A.: *The Resolution Calculus,* Springer (1997)

[Letz 92] Letz, R., Schumann, J., Bayerl, S., and Bibel, W.: SETHEO: A High-Performance Theorem Prover, *J. of Automated Reasoning,* Vol. 8, No. 2, pp. 183-212 (1992)

[Letz 94] Letz, R., Mayr, K., and Goller, C.: Controlled Integration of the Cut Rule into Connection Tableau Calculi, *J. of Automated Reasoning,* Vol. 13, pp. 297-337 (1994)

[Manthey 88] Manthey, R. and Bry, F.: SATCHMO: a theorem prover implemented in Prolog, in Lusk, E. and Overbeek, R. eds., *9th Conference on Automated Deduction,* Vol. 310 of *LNCS,* pp. 415-434, Springer-Verlag (1988)

[Niemelä 96] Niemelä, I.: A Tableau Calculus for Minimal Model Reasoning, in *Theorem Proving with Analytic Tableaux and Related Methods,* Vol. 1071 of *LNAI,* pp. 278-294, Springer (1996)

[丹治 99] 丹治信春: タブローの方法による論理学入門, 朝倉書店 (1999)

[Oppacher 88] Oppacher, F. and Suen, E.: HARP: A Tableau-Based Theorem Prover, *J. of Automated Reasoning,* Vol. 4, pp. 69-100 (1988)

[白井 97] 白井, 長谷川: モデル生成型定理証明システムによる制約充足問題の解決とその並列化, 電子情報通信学会論文誌, Vol. J80-D-II, No. 1, pp. 224-236 (1997)

2001年7月9日 受理

著者紹介

長谷川 隆三 (正会員) は, 前掲 (Vol. 16, No. 2, p. 327) 参照.

藤田 博 (正会員) は, 前掲 (Vol. 16, No. 2, p. 327) 参照.

越村 三幸 (正会員) は, 前掲 (Vol. 16, No. 2, p. 327) 参照.