分岐補題の導入による極小モデルの効率的生成法

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

藤田, 博
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

越村, 三幸
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

https://doi.org/10.15017/1513728

出版情報：九州大学大学院システム情報科学紀要. 4 (2), pp.145-150, 1999-09-24. 九州大学大学院システム情報科学研究院

九州大学大学院
システム情報科学研究科報告
第4巻 第2号 平成11年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.4, No.2, September 1999

分岐補題の導入による極小モデルの効率的生成法

長谷川隆三*・藤田 博*・越村三幸*

An Efficient Method to Generate Minimal Models by Introducing Splitting Lemmas

Ryuzo HASEGAWA, Hiroshi FUJITA and Miyuki KOSHIMURA

(Received June 21, 1999)

Abstract: An efficient method for minimal model generation is presented. The extended MGTP, called MM-MGTP, was implemented using the method so that every model it generates is guaranteed to be minimal. The method employs *splitting assumption* which is equivalent to Bry's *complement splitting rule* as a basic mechanism. Moreover, we introduce a new concept called *splitting lemma* which can avoid performing unnecessary tests on minimal models and prune branches leading to non-minimal models. Experimental results with MM-MGTP show remarkable speedup compared to Bry's MM-SATCHMO.

Keywords: Minimal model, Model generation, Automated theorem proving, Proof theory

1. はじめに

自動定理証明，論理プログラミング，演繹データベース等の分野においては，極小モデルの概念が重要である．プログラム検証，故障診断，仮説推論などの応用上も実際に極小モデルの生成が必要になる場合が多い．

モデル生成法に基づく定理証明系はすべての極小モデルを生成しうるが，非極小モデルも生成する．我々が研究開発を進めているMGTPのベースとなったSATCHMOの開発グループが，最近，極小モデルを生成する決定手続きを与え，MM-SATCHMO[1]を提案した．本手続きは，"分岐仮定"と"モデル制約"の導入により，非極小モデルを棄却する．しかしながら，この方法だけでは非極小モデルを導く冗長な部分証明木を重複して生成したり，不要な極小性テストを行うなどの冗長な計算が含まれるため，著しく効率を低めてしまう場合があることが判明した．我々はこの問題を解決するために，"分岐補題"なるさらに協力な概念を導入することに成功した．

一般に，モデルMの極小性を知るには他のどのモデルからも包摂されないことを確認する必要がある．しかし，証明木において他のモデルとの間に"委託関係"(Mの要素を用いて閉じる部分証明木)がなければMの極小性が保証されるので，極小性テストを省くことができる．さらに，非極小モデルとなることが判る部分証明木を刈り込むことによって探索空間を狭めることができる．これらは，本稿で提案する分岐補題により実現可能になる．Java版CMGTP[2]をベースに，分岐補題を導入した極小モデル生成器MM-MGTPを実装し，極めて良好な結果を得た．

平成11年6月21日受付

* 知能システム学専攻

2. モデル生成型定理証明系 CMGTP

CMGTP[3]はモデル生成法に基づく一階述語論理の定理証明系である．$A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_m$ なる含意式表現の節の集合に対し，**Fig.1**のような手続きでモデルの構成を試みる．ここで，A_i, B_jはリテラル，→の左辺を前件，右辺を後件という．また，$m = 0$を負節，$m = 1$をホーン節，$m \geq 2$を非ホーン節という．

連言照合関数$CJM(u, M)$は，モデル候補Mとモデル拡張候補アトムuで充足される前件をもつ節の具体例を探し，ホーン節の後件の集合と非ホーン節の後件の集合の対を返す．$Simp\&Subsump(D, M)$では**Fig.2 (b)**に示す選言縮約と包摂テストを実行する．

モデル候補の棄却条件は，1) CJM関数において負節に対して連言照合が成功する，2) **Fig.2 (a)**の単位反駁が導かれる，3) **Fig.2 (b)**の選言縮約によってD中のある選言が空となる，場合に成立する．

節集合が充足不能のときは，すべてのモデル候補を棄却して有限終了する．一方，節集合が充足可能な場合，無限個の要素をもつモデルが存在しない限り，すべてのモデルを生成することができる．

CMGTP手続きに従い構成されるモデル候補集合$\mathcal{M}$は，木で表現することができる．これを"MG木"と呼ぶ．MG木の根から一つの葉に至る経路上の節点をラベル付けするアトムの集合が一つのモデル(候補)を表す．×印の葉は対応するモデル候補が棄却されたことを，○印の葉は対応するモデル候補がモデルであることを示す．

例えば，**Fig.3**の節集合S0に対する手続き終了後のMG木は同図右のようになる．

```
M0 = ⟨φ, U0, D0⟩; 𝓜 = {M0};
/* U0: (Consequents of Horn clauses) */
/* D0: (Consequents of non-Horn clauses) */
L1: while (𝓜 ≠ φ) do { M = ⟨M, U, D⟩ ∈ 𝓜;
  while (U ≠ φ) { U = U \ {u ∈ U};
    if (u ∉ M) /* Unit subsumption */ {
      M = M ∪ {u}; ⟨U′, D′⟩ = CJM(u, M);
      U = U ∪ U′; D = D ∪ D′;
      D = Simp&Subsump(D, M);
      when ( Model candidate rejected ) {
        𝓜 = 𝓜 \ M; continue L1; } } }
  if (D = φ) { output MODEL(M);
    𝓜 = 𝓜 \ M; continue L1; }
  else { d = (d1 ∨ ... ∨ dm) ∈ D;
    Mk = ⟨M, U ∪ {dk}, D \ {d}⟩;
    𝓜 = (𝓜 \ M) ∪ {Mk}(1 ≤ k ≤ m); } }
if (/* No model found */) output UNSAT;
```

Fig.1 CMGTP procedure

$$\frac{a \in M \quad \neg a \in M}{\perp} \qquad \frac{a(\neg a) \in M \quad \neg a(a) \vee \mathcal{C} \in D}{\mathcal{C}}$$

(a) Unit refutation (b) Disj-simplification

Fig.2 Inference rules in CMGTP

$S0 =$
$\{ \rightarrow a \vee b.$
$a \rightarrow .$
$b \rightarrow c. \}$

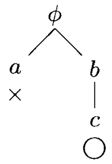

Fig.3 Clause set S0 and its MG-tree

3. 極小モデル生成

CMGTPは非極小モデルを生成することがある．例えば，**Fig.4** の節集合S1のMG木は同図右のようになる．すなわち，$\{a, b\}$と$\{b\}$の二つのモデルが得られるが，前者は“後者に包摂される(後者を包含する)”ので非極小モデルである．MG木の葉に付された$\odot(\otimes)$は，対応するモデル候補が極小モデル(非極小モデル)であることを示す．

3.1 分岐仮定

節集合S1の節(1)を**Fig.5** のように(3)†1で置き換えると，生成されるMG木は同図右のようになる．ここで，$[\neg b]$を“分岐仮定(*splitting assumption*, SA)”と呼ぶ．この場合，モデルは$\{b\}$のみとなる．一般に，$Ante \rightarrow B_1 \vee B_2 \vee \ldots \vee B_n$ に対しては分岐仮定を，$Ante \rightarrow B_1 \wedge [\neg B_2] \wedge \ldots \wedge [\neg B_n] \vee B_2 \wedge [\neg B_3] \wedge \ldots \wedge [\neg B_n] \vee \ldots \vee B_n$ のように付加する．これは，Bry[1]らの“相補分割

†1 (3)は，厳密には節ではなく，$a, [\neg b]$ と b の枝に分岐することを意味する．

$S1 =$
$\{ \rightarrow a \vee b. \quad (1)$
$a \rightarrow b. \quad (2) \}$

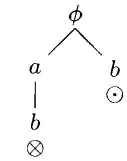

Fig.4 Clause set S1 and its MG-tree

$S2 =$
$\{ \rightarrow a \wedge [\neg b] \vee b. \quad (3)$
$a \rightarrow b. \quad (4) \}$

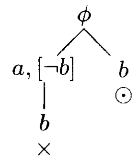

Fig.5 MG-tree for clause set S2 with CA

$S3 =$
$\{ \rightarrow a \wedge [\neg b] \vee b. \quad (5)$
$b \rightarrow a. \quad (6) \}$

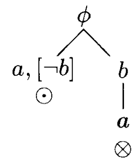

Fig.6 MG-tree for clause set S3 with CA

(*complement splitting*)”規則の適用に相当している．CMGTPにおいては分岐仮定は節の書換えのみで実現可能である．分岐仮定を用いれば，MG木の最左枝として得られるモデルは極小モデルとなる．しかし，最左でないモデルについては極小モデルとは限らない．例えば，**Fig.6** の節集合S3に対するMG木は同図右のようになり，$\{a\}$なる極小モデル(分岐仮定$[\neg b]$はモデル要素からは除外する．)の他に，非極小モデル$\{b, a\}$も得られてしまう．

3.2 モデル制約

Bry[1]らは，非最左枝として得られるモデルの極小性を保証するために，“モデル制約(*model constraint*, MC)”を導入した．これは，すでに確定した極小モデル$\{e_1, e_2, \ldots, e_m\}$について，$e_1 \wedge e_2 \wedge \ldots \wedge e_m \rightarrow$ なる負節を節集合に加えるというものである．**Fig.6** の例では，極小モデル$\{a\}$が得られた時点で，負節$a \rightarrow$がS3に加えられる．これにより，非極小モデル$\{b, a\}$は棄却される．モデル制約の実現にはCMGTP手続きの変更が必要となる．以後，極小モデル生成手続きを持つCMGTPをMM-MGTPと呼ぶ．

MM-MGTPの実現に際しては，Bryらのように負節を動的に加える代りに，得られた極小モデルをMG木としてそのまま保持し，MG木を辿ることにより極小性テストを行う方式を採った．本方式を“モデル記憶方式”と呼ぶ．しかし，いずれにしてもモデル制約テストのためには得られた各極小モデルを記憶する必要があるため，記憶量の増大が問題となる．

$S4 =$

$$\{ \rightarrow a \wedge [\neg b] \vee b. \quad (7)$$
$$a \rightarrow b. \quad (8)$$
$$b \rightarrow a. \quad (9)\ \}$$

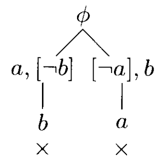

Fig.7 MG-tree for clause set S4 with CA and CL

3.3 モデル再計算

上記の問題を解決するため，Niemelä[4]は，分岐仮定と類似の*groundedness*テストを導入している．このテストは，モデル$M = \{e_1, \ldots, e_n\}$が得られたとき，$\forall a \in M\ S \cup \overline{M} \models a$を判定する．ただし，$S$は与えられた節集合，$\overline{M} = \{\neg b \mid b \notin M\}$である．実際には，$S \cup \overline{M}$に$\{\leftarrow e_1 \wedge \ldots \wedge e_n\}$を加えてタブローを再計算し，充足不能性をテストする．充足不能ならばMは極小，充足可能ならば非極小である．

MM-MGTPにおいては，モデルMが得られたとき，分岐仮定の下で$S \cup \overline{M}$に対してモデル生成を再度実行する．この過程で少なくとも一つのモデル(M自身)が生成されることに注意．したがって，この再実行はモデルの単解探索でよい．再実行時に生成されたモデルがMに一致する場合は，Mが極小であることを意味する．そうでない場合は，Mに真に包含されるモデルが存在したことになり，Mは極小でないと判定される．なぜならば，$\overline{M}$の前提により，Mより大きなモデルは生成され得ないからである．これを，“モデル再計算方式”と呼ぶ．この方式は，本質的にNiemeläの方法と同等であり，Mを包摂する極小モデルが存在すれば，MG木においてMの左側に見つかる．

4. 分 岐 補 題

ここまで分岐仮定は“非対称的に”付加するものとした．これを対称的に加えた場合，一般に極小モデル生成に関して不完全となる．例えば，**Fig.7**の節集合S4は極小モデル$\{a, b\}$をもつが，節(7)を分岐仮定が対称的に付加された$\rightarrow a \wedge [\neg b] \vee [\neg a] \wedge b$ (10) で置き換えると，同図右のようにモデルは一つも得られない．これは，aの証明で$[\neg b]$を使用(即ち，aの証明をbに委託)し，bの証明で$[\neg a]$を使用(bの証明をaに委託)しているためである．

しかし，前述の節集合S3では，節(5)を節(10) で置き換えても**Fig.6**と同様に極小モデル$\{a\}$が得られ，さらに非極小モデル$\{b, a\}$を$[\neg a]$によって棄却することができる．これは，aの証明で$[\neg b]$を使用していないので，相互委託が生じないからである．このような$\neg a$を“分岐補題(*splitting lemma*, SL)”と呼び，以後$\{\neg a\}$と表す．

極小モデル生成の完全性を保証する分岐補題の付加の仕方は一般に次のようになる．非対称な分岐仮定を付加した節$Ante \rightarrow B_1 \wedge [\neg B_2] \wedge \ldots \wedge [\neg B_n] \vee B_2 \wedge [\neg B_3] \wedge \ldots \wedge [\neg B_n] \vee \ldots \vee B_n$ のB_iの分岐から下に生成されるMG木において，分岐仮定$[\neg B_j]$ $(i+1 \leq j \leq n)$を用いてモデル棄却または選言簡約化が行われない場合，$\{\neg B_i\}$を分岐補題としてB_jの各分岐に加える．

$S5 =$

$$\{ \rightarrow a \wedge [\neg b] \vee b. \quad (11)$$
$$a \rightarrow b \vee c. \quad (12)$$
$$b \rightarrow a. \quad (13)\ \}$$

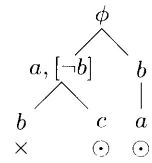

Fig.8 MG-tree for clause set S5 with CA

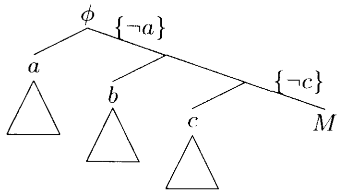

Fig.9 Partial minimality test

$S6 =$

$$\{ \rightarrow a \wedge [\neg b] \vee b. \quad (14)$$
$$\rightarrow c \wedge [\neg d] \vee d. \quad (15)\ \}$$

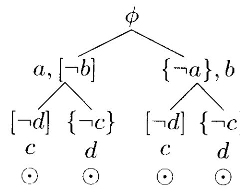

Fig.10 MG-tree for clause set S6 with CA and CL

分岐補題を生成できない場合として，**Fig.8**の節集合S5を考えよう．同図右に示すように，aの枝では$[\neg b]$が使用されているため分岐補題$\{\neg a\}$を生成できない．したがって，aの弟分岐bの下で得られたモデル$\{b, a\}$については，極小モデル$\{a, c\}$に対する極小性テストが省けない．ただし，極小性テストは分岐補題を生成しえなかった分岐の下に対してのみ行えばよい．例えば，**Fig.9**のように$\{\neg a, \neg c\}$の分岐補題の下でモデルMが得られた場合，Mの極小性テストは分岐補題を生成しえなかったbの下に対してのみ行う．一般に，Mの経路上の各分岐節点において全兄節点から分岐補題が生成されている場合は，極性性テストを省くことができる．例えば，**Fig.10**の節集合S6ではすべての分岐で分岐補題を生成できるので，極小性テストを全く行わずに，4つの極小モデルが得られる．

4.1 枝刈り

モデル候補Mが極小性テストにより棄却される場合，Mより右のMG木において不要なモデル探索枝を刈れる場合がある．例えば，**Fig.11**の節集合S7でモデル$M_d : \{b, a, c, d\}$が得られたとき，極小性テストの結果，極小モデル$\{a, c\}$により包摂されることが判り，棄却された状況を考える．この後，モデル$M_e : \{b, a, c, e\}, M_f : \{b, a, c, f\}$が順次得られるが，ともに極小性テストで棄却される．ここで，M_e, M_fの共通部

$S7 =$

$\{ \rightarrow a, [\neg b] \vee b.$ (16)

$a \rightarrow b \vee c.$ (17)

$b \rightarrow a.$ (18)

$b \wedge a \rightarrow c.$ (19)

$b \wedge a \wedge c \rightarrow$

$d \vee e \vee f.$ (20) $\}$

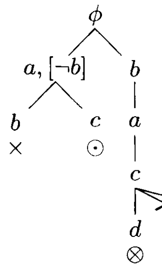

Fig.11　MG-tree for clause set S7

分$\{b, a, c\}$がすでに極小モデル$\{a, c\}$により包摂されることを知れば，M_e, M_fを展開する前に棄却できる．すなわち，M_dの棄却と同時にc以下のすべての枝を刈り取ってよい．一般に，棄却されるモデルのパス上で，これを包摂した極小モデルの要素をラベルにもつ節点のうち葉から最も近い節点以下を刈り取ればよい．

5.　極小モデル生成の健全性と完全性

定理 1　**Fig.12 (a)**に示すように，あるモデル拡張によって生じた分岐を$L_1, L_2, \ldots, L_m$とし，L_iの下の部分証明木をP_i，L_jの下の部分証明木をP_j $(1 \leq i, j \leq m\, ; i \neq j)$とする．また，$P_j$の中に$L_i$が出現しており，この$L_i$を根とする部分証明木を$P_{L_i}$とする(**Fig.12 (b)**)．

このとき，○印が付されたP_{L_i}の任意の葉(これに対応するモデルをMとする)に対して，○印が付されたP_iのある葉(これに対応するモデルをM'とする)が存在し，$M' \subseteq M$が成り立つ．□

証明：P_iの根のモデル拡張が**Fig.12 (c)**のようであったとする．このモデル拡張を行った節を$\Gamma_1 \rightarrow L_1^1; \ldots; L_{m_1}^1$とすると$M$はモデルなので，$M \models \Gamma_1$かつ$M \models L_1^1; \ldots; L_{m_1}^1$が成り立つ．したがって，$\exists i (1 \leq i \leq m_1) L_i^1 \in M$が成り立つ．今$L_i^1 \in M$を満たすような$i$を$k_1$とし，$L_{k_1}^1$を根とする部分証明木の根のモデル拡張を行った節を$\Gamma_2 \rightarrow L_1^2; \ldots; L_{m_2}^2$とする．上と同様に考えて，$\exists i (1 \leq i \leq m_2) L_i^2 \in M$が成り立つ．

以降，同様な手続きを行っていくと$L_{k_1}^1, L_{k_2}^2, \ldots$というラベルの系列が得られる．この系列は○で終了し，×で終了することはない．この○印が付された終端に対応するモデルをM'とすると$M' \subseteq M$が成り立つ．◇

定義 1 (証明木の委託関係)　証明木の委託関係とは，兄弟節点間の厳密な半順序関係$\prec$のことである．($N_1 \prec N_2$は，N_2の下のあるモデル探索をN_1の下のモデル探索に委託したことを表す．)□

定義 2 (証明委託)　証明委託とは，次のような処理をいう．今，証明途中の証明木において，開である葉N_1とその祖先節点N_3の兄弟の節点N_2が同一のL_1でラベル付けされているものとする(**Fig.13**参照)．また，$N_3 \not\prec N_2$であるとする．このとき，N_1から枝を下方に伸ばし，その

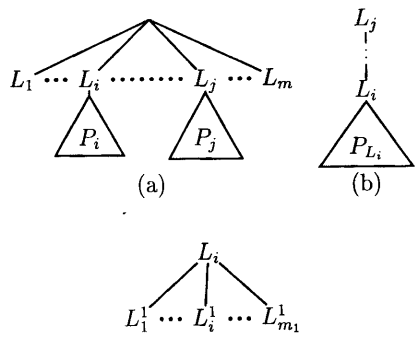

Fig.12　MG-trees explaining Theorem 1

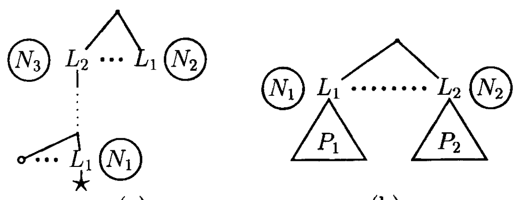

Fig.13　MG-trees explaining Theorem 2

先端節点に★の印を付すとともに，委託関係$\prec$を$N_2 \prec N_3$を加えて修正する．★印は，そこから下のモデルの探索を他に委託したことを表す．□

系 1　通常のモデル生成で求まる極小モデルは，証明委託を行うモデル生成によっても求まる．□

証明：定理1 より明らか．◇

定理 2　証明委託を行うモデル生成後に，節点N_1とN_2の間に$N_2 \not\prec N_1$が成り立っているものとする．このときN_1の下の部分証明木P_1の葉に対応するモデルM_1とN_2の下の部分証明木P_2の葉に対応するモデルM_2について，$M_1 \subset M_2$が成り立つことはない．□

証明：N_1はL_1で，N_2はL_2でラベル付けされているものとする(**Fig.13**参照)．

(1) $N_1 \prec N_2$の場合，委託の性質により$L_1 \notin M_2$．一方，$L_1 \in M_1$．したがって，$M_1 \not\subset M_2$．

(2) N_1とN_2の間に委託関係$\prec$がない場合，委託の性質により$L_1 \notin M_2$．よって，$M_1 \not\subset M_2$．◇

定理2 により，モデル制約による極小性判定は，$N_2 \prec N_1$を満たすような節点の下で求まるモデルM_1とM_2に対してのみ，$M_1 \subset M_2$であるかの判定を行えば十分であることがわかる．

定義 3 (制限されたモデル制約テスト)　二つの節点N_1とN_2について，P_iはN_iを根とする部分証明木，M_iは○印が付されたP_iの任意の葉に対応するモデルとする$(i = 1, 2)$．$N_2 \prec N_1$を満たすような節点間についてのみ$M_1 \subset M_2$かどうかのテストを行うことを制限されたモデル制約テストという．□

定理 3 制限されたモデル制約テストと証明委託を行うモデル生成は，極小モデル生成に関して健全(極小モデルしか計算しない)であり完全(すべての極小モデルを計算する)である． □

証明：定理2 により健全性がいえる．また，通常のモデル生成ですべての極小モデルが求まるので系1 から完全性がいえる． ◇

前節まで述べた分岐仮定，モデル制約，分岐補題を用いるモデル生成は，制限されたモデル制約テストと証明委託を行うモデル生成の性質を満たすので，次の系が成り立つ．

系 2 分岐仮定，モデル制約，分岐補題を用いるモデル生成は，極小モデル生成に関して健全かつ完全である． □

6. 実 験 評 価

MM-MGTP と他のシステム(MM-SATCHMO, CMGTP)との比較評価をおこなった．MM-MGTPに関しては，分岐補題の効果を調べるため，モデル再計算方式(Rcmp)とその分岐補題つき(Rcmp+SL)，およびモデル記憶方式(Mchk)とその分岐補題つき(Mchk+SL) の4通りを比較した．MM-MGTP とCMGTP はJavaで，MM-SATCHMO はECL^{i}PS^e Prologで実装されており，計測はすべてSun Ultra10(333MHz,128MB)上で行った．

評価に用いた例題は，以下のとおりである．

ex1

$$S_N = \{\rightarrow p_{i1} \vee p_{i2} \vee p_{i3} \vee p_{i4} \vee p_{i5} \vee p_{i6} \vee p_{i7} \vee p_{i8} \vee p_{i9} \vee p_{i10} \mid i = 1, \ldots, N\}$$

文献1)における最悪ケースの例題 $A(5,10)$ で，MG木は10分岐の平衡木であり，各分岐で分岐補題を生成できるので，極小性テストが不要となる．

ex2

$$S_N = \{p_{i+1} \rightarrow p_i \vee q_i \vee r_i,\ r_i \rightarrow q_i,\ q_i \rightarrow p_i \mid i = 1, \ldots, N\} \cup \{\rightarrow p_N\}$$

MG木は，右に行くほど深く葉数も増大する非平衡木である．最左枝の1本のみが極小モデルであり，その他のモデルはすべてこれに包摂される．分岐補題を用いれば，これらはすべて棄却できる．

ex3

$$T_1 = \{\rightarrow a_1 \vee b_1,\ a_1 \rightarrow b_1,\ b_1 \rightarrow a_2 \vee b_2,\ a_2 \rightarrow b_2 \vee d_1\}$$

$$T_2 = \{b_2 \rightarrow a_3 \vee b_3,\ a_3 \rightarrow a_2 \vee c_2,\ a_3 \wedge a_2 \rightarrow b_3 \vee d_2,\ a_3 \wedge c_2 \rightarrow b_3 \vee d_2\}$$

$$T_j = \{b_j \rightarrow a_{j+1} \vee b_{j+1},\ a_{j+1} \rightarrow a_j \vee c_j,\ c_j \rightarrow a_{j-1} \vee c_{j-1},\ a_{j+1} \wedge a_2 \rightarrow b_{j+1} \vee d_j,\ a_{j+1} \wedge c_2 \rightarrow b_{j+1} \vee d_j\}\ (j \geq 3)$$

$$S_N = \bigcup_{i=1}^{N} T_i$$

MG木は，ex2と同様に右重の非平衡木である．各モデルの経路上に分岐補題を生成しない分岐点が存在するので，極小性テストが必要となるが，結局包摂されない．

syn9-1

定理証明ベンチマーク集TPTP[5]から選んだ例題で，充足不能問題である．

channel

チャネルルーティングの例題で，CMGTPの負アトムによる制約伝播機構が不可欠となる．CMGTPを用いてもすべて極小モデルが得られる．

Table–1 に性能比較を示す．MM-SATCHMOは，メモリオーバフローのため計測不能となる場合が多い．結果が得られたex1(N=5)，ex3では，実行時間はMM-MGTPの最速の場合に比べると5桁以上遅い．**Table–1** には示していないが，文献1)に示された例題 $D(4,5,1)$では，MM-SATCHMOで0.25秒，Niemeläシステムで2秒，分岐補題つきモデル記憶方式のMM-MGTPで0.015秒であった．例題$D(4,5,4)$では，それぞれ0.09秒，0.5秒，0.013秒であった．

一方，CMGTPと比較すると，ex2, ex3のように非極小モデルの数が多い場合ほど，MM-MGTPの実行時間は短縮される．また，非極小モデルが生成されないex1, syn9-1, channelのような問題でも，分岐補題つきの場合にはほとんどオーバヘッドが見られない．

再計算方式(Rcmp)とモデル記憶方式(Mchk)を比較すると，再計算オーバヘッドによる実行時間の増加は，項メモリを用いないex1からex3までの例題では2～6倍程度である(ただし，ex1(N=7)を除く)．channelにおいては約10倍の開きがあるが，これは項メモリのアクセスオーバヘッドが大きいことによる．

次に，再計算方式とモデル記憶方式について，分岐補題の有無で比較する．ex1は分岐補題により極小性テストを省くことができるため，再計算方式では分岐補題つきの方が約9～13倍速い．一方，モデル記憶方式では分岐補題つきの方が約2倍速い．

分岐補題つきの再計算方式(Rcmp+SL)とモデル記憶方式(Mchk+SL)を比較すると，極小性テストが省けないex3では，モデル記憶方式の方が約5倍速い．しかし，極小性テストが不要なex1(N=5)では差異が生じないはずであるが，約2倍遅くなっている．これは，モデル記憶方式の場合，生成されたモデルをすべて記憶するので，GC(ゴミ集め)が多発したためと思われる．また，ex1(N=7)ではメモリオーバフローが生じ(最大ヒープサイズ=67MBで走行)，計算不能となる．

7. 関 連 研 究

モデル記憶方式のMM-MGTPはMM-SATCHMOの改良を狙ったものであり，基本的には文献1)に示された相

Table-1 Performance comparison

Problem	Rcmp+SL	Mchk+SL	Rcmp	Mchk	MM-SATCHMO	CMGTP
ex1 (N=5)	0.719 100000 0	1.342 100000 0	6.569 100000 0	2.282 100000 0	8869.650 100000 0	0.555 100000 0
ex1 (N=7)	71.726 10000000 0	OM (> 144) — —	899.473 10000000 0	OM (> 115) — —	OM (> 40523) — —	55.064 10000000 0
ex2 (N=14)	0.015 1 26	0.015 1 26	254.020 1 1594322	40.631 1 1594322	1107.360 1 1594322	26.778 1594323 0
ex3 (N=16)	54.763 65536 1	11.304 65536 1	51.286 65536 1	10.851 65536 1	OM (>2798) — —	1520.686 86093442 0
syn9-1	0.296 0 19683	0.283 0 19683	0.259 0 19683	0.252 0 19683	TO (> 61200) — —	0.273 0 19683
channel	8.582 51922 78	10.127 51922 78	101.769 51922 78	10.951 51922 78	NA — —	8.321 51922 78

top: time (sec), middle: No. of models, bottom: No. of failed branches.
OM: Out of memory, TO: Time out, NA: Not available due to lack of constraint handling

補分割規則とモデル制約による極小モデル生成法に基づいている．しかし，1)では，本稿で提案した分岐補題を導入していないため，非極小モデルを導く冗長な枝を刈り込むことができず，また，本来必要のないモデル制約に対する極小性テストを行ってしまう．一方，MM-MGTPではかかる冗長計算は除去されている．

さらに，1)では，得られたモデル制約を負節として記憶し，極小性テストを負節に対する連言照合により行っているため，実行効率が著しく悪い．これに対して，モデル記憶方式のMM-MGTPでは，モデル制約はMG木として記憶するので共通経路の共有化が図られ，しかも極小性テストは分岐補題を生成できなかった枝に対してのみ部分的に行うので効率的である．

上記のMM-SATCHMOおよびモデル記憶方式のMM-MGTPは，一種のmemoizationであるモデル制約に基づいているため，必要な記憶域は最悪の場合指数的に増大する．これを避けるアプローチとして，Niemelä[4]により，命題論理の極小モデル生成手続きがタブロー法の枠組みで提案されている．文献4)では，分岐仮定と類似のgroundednessテストを導入することにより，以前に生成された極小モデルに包摂されるか否かのテストをせずに非極小モデルを棄却できる．しかし，モデル制約によるよりも複雑なgroundednessテストを繰り返す（モデルが得られる度に，そのモデルに含まれないアトムに否定を付与し，再度節集合の充足可能性をテストする）他，同じ極小モデルが何度も生々されることを抑止できない．また，1)と同様に，分岐補題を導入していないため，無駄なテストが行われてしまう．

再計算方式のMM-MGTPは，モデル制約を記憶しなくてよいという4)の利点を活かしたものである．しかし，4)とは異なり，動的に負節を生成する必要がなく，これに係わる連言照合も省くことができる．さらに，本方式においても分岐補題を活用できるので，高価なgroundednessテストを必要極小限に留めることができる．

8. ま と め

本稿では，分岐補題を利用して効率的に極小モデルを計算する方法を提案した．極小モデルを計算する方法として，既に相補分割規則とモデル制約に基づくBryらの方法とgroundednessテストに基づくNiemeläの方法が提案されているが，いずれにおいても，極小性テストに冗長な計算が含まれている．これらの冗長性は分岐補題を用いると除去できることを示し，実験評価によって数桁以上の効率改善が得られることを確認した．

しかしながら，分岐補題を適用できない場合には高価な極小性テストに頼らざるをえない．したがって，分岐補題の適用範囲の拡大による極小性テストの対象のさらなる限定化，証明木変換技法に基づく極小モデル集合の縮約などの方法を確立することが今後の課題である．

参 考 文 献

1) Bry, F. and Yahya, A.: "Minimal Model Generation with Positive Unit Hyper-Resolution Tableaux," *LNAI 1071,* (1996) 143–159.
2) 長谷川 隆三, 藤田 博: "制約問題を解くためのモデル生成型定理証明系の新実装法," 九州大学大学院システム情報科学研究科報告, Vol. 4, No. 1, (1999) 57–62.
3) 白井 康之, 長谷川 隆三: "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, (1997) 224–236.
4) Niemelä, I.: "A Tableaux Calculus for Minimal Model Reasoning," *LNAI 1071,* (1996) 278–294.
5) Sutcliffe, G., Suttner, C. and Yemenis, T.: "The TPTP Problem Library," *Proc. CADE-12,* (1994) 252–266.