ICOT Technical Report: TR-665

TR-665

モデル生成型証明器上の様相命題タブロ

越村　三幸、長谷川　隆三

July, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

モデル生成型証明器上の様相命題タブロ

Modal Propositional Tableaux in a Model Generation Theorem Prover

越村 三幸
Miyuki Koshimura
東芝情報システム(株)
Toshiba Information Systems (Japan)

長谷川 隆三
Ryuzo Hasegawa
新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology

概要

モデル生成型証明器上のタブロ証明器について述べる. タブロ法に基づく証明法とモデル生成法に基づく証明法の類似性に着目し, モデル生成法に基づく自動証明器 MGTP(Model Generation Theorem Prover) 上に, タブロ法による様相命題論理の自動証明器を作成した. 具体的には, MGTP への入力節集合をプログラミング言語とみなし, 証明器をプログラムすることにより MGTP 上に別種の自動証明器を作成した. この二段階証明器作成法では, 証明器の対象領域の性質 (推論規則) は入力節集合として表現され, 推論機構は MGTP のプログラムとして表現される. これにより対象領域の性質と推論機構の記述が分離され, 見通しのよい自動証明器の作成が可能となる.

1 はじめに

自動証明器は, 定理を与えるとその証明図を定められた推論規則を用いて構成しようと試みる. そして, その推論規則は基礎となる論理体系によって異なる. 例えば, 導出原理では「導出」と呼ばれる推論規則を用いて証明図を構成するし, タブロ法では「分解規則」を用いて証明図を構成する.

自動証明器をプログラムの視点から眺めると, 一般的に, 定理は入力データとして与えられ, 証明図構成法はプログラムとして記述されている. この場合, 証明図の構成は入力データに基づいて行なわれる. 見方を変えると, 入力データとしての定理は証明図の構成方法を示す仕様と見なすこともできる. このような見方により, 従来プログラムとして記述していた証明図構成に関わる部分を, 入力データとして記述することが可能となる.

本論文では, この見方に基づいた証明器について報告する. 具体的にはタブロ法 [5] の分解規則を自動証明器 MGTP[3] の入力データとして記述することにより, MGTP 上にタブロ証明器の構成する.

証明器の上に別の証明器を作成するこの方法では, Prolog インタプリタの上に Prolog インタプリタを実現した時のような, いわゆるインタプリテーションオーバヘッドによる低速化が懸念される. しかし, MGTP では入力データは予めコンパイルされるために, このオーバヘッドは極めて少ない.

MGTP は, モデル生成法に基づく一階述語論理のための並列定理証明器であり KL1[6] により実現されている. MGTP は入力節集合のモデルをボトムアップ的に生成しようと試みる. モデルの生成に成功すれば入力節集合は充足可能であることが分かり, 生成に失敗すれば充足不可能であることが分かる.

$$apply(MGTP, ASetOfClauses) = \begin{cases} satisfiable \\ unsatisfiable \end{cases}$$

一方タブロ法では, 入力論理式のモデルを論理式の分解により生成しようと試みる.

モデル生成の観点からの MGTP とタブロ法の証明図の類似性に着目し, タブロ法の分解規則を節形式で表現することにより, MGTP 上にタブロ証明器を容易に作成することができる.

$$apply(MGTP, TableauxProver(Formula)) = \begin{cases} satisfiable \\ unsatisfiable \end{cases}$$

実際タブロ法の 閉条件 / 入力論理式 / 分解規則は, それぞれ MGTP の負節 / 正節 / その他の節 として自然に記述される. これはモデル生成法とタブロ法の相性がいいことを示している.

我々は, 実際に幾つかの非標準命題論理の証明器を MGTP 上に作成した. これらの証明器は様相タブロ法 [1, 2] に基づいている. 次節以降で MGTP と様相タブロ法の簡単な説明を行った後, MGTP 上の証明器の作成法を示す. また, 時相論理の例題による性能測定についても述べる.

2 MGTP

MGTP はモデル生成法に基づき,与えられた節集合のモデルを生成しようと試みる. 本節では MGTP の実現のための詳細には立ち入らず,モデル生成法の概略と MGTP の特徴について述べる.

2.1 モデル生成法

本論文を通じて節は次のように含意式の形で表現される.

$$A_1, A_2, \ldots, A_n \rightarrow C_1; C_2; \ldots; C_m$$

または,

$$A_1, A_2, \ldots, A_n \rightarrow C_1, C_2, \ldots, C_m$$

ここで,$A_i(1 \leq i \leq n)$ および $C_j(1 \leq j \leq m)$ は原子論理式である. → の左側を前件部,右側を後件部という. 前件部は $A_1, A_2, \ldots, A_n$ の連言(','), 後件部は $C_1, C_2, \ldots, C_m$ の選言(';') または連言(',') である. 節は前件部が $true(n = 0)$ のとき正節, 後件部が $false(m = 0)$ のとき負節と呼ばれる.

モデル生成法には次の二つの規則がある.

- モデル拡張規則: ある節 $A \rightarrow C$ において, 置換 σ のもとに $A\sigma$ が M で充足されており,$C\sigma$ が M で充足されてないとき, モデル M に $C\sigma$ を加えてこれを拡張する.
- モデル棄却規則: ある負節で前件 $A\sigma$ が M で充足されるとき, M を棄却する.

ここで,$A\sigma$ を得る過程を前件のモデル要素との連言照合という. 正節では前件 ($true$) がどんなモデルでも充足されることに注意されたい.

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ということを行なうものである. もし節集合が充足可能ならば, モデルが見つかるはずである. また, あらゆるモデルの可能性を探索した上で結局一つのモデルも構成できないとき, その節集合が充足不可能であることがわかる.

例として次の節集合 S1 を考える.

$$\begin{aligned}
C1 &: \quad p(X), s(X) \rightarrow false. \\
C2 &: \quad q(X), s(Y) \rightarrow false. \\
C3 &: \quad q(X) \rightarrow s(f(X)). \\
C4 &: \quad r(X) \rightarrow s(X). \\
C5 &: \quad p(X) \rightarrow q(X); r(X). \\
C6 &: \quad true \rightarrow p(a); q(b).
\end{aligned}$$

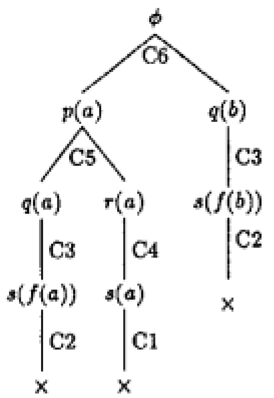

図 1: S1 に対する証明図

S1 問題に対する証明木を図 1に示す. まず, 空集合 $M_0 = \phi$ から始める. $C6$ にモデル拡張規則を適用することにより, M_0 は $M_1 = \{p(a)\}$ と $M_2 = \{q(b)\}$ に場合分けされる. 次に,$C5$ によって,M_1 は $M_3 = \{p(a), q(a)\}$ と $M_4 = \{p(a), r(a)\}$ に場合分けされる. さらに,$C3$ によって M_3 は $M_5 = \{p(a), q(a), s(f(a))\}$ に拡張される. さて,M_5 においてモデル棄却規則が $C2$ に適用できて, M_5 は棄却され, このケースはここで終りとなる. 一方,M_4 は $C4$ によって $M_6 = \{p(a), r(a), s(a)\}$ に拡張されるが,$C1$ によって棄却される. 同様に,M_2 は $C3$ によって $M_7 = \{q(b), s(f(b))\}$ に拡張された後,$C2$ によって棄却される. 今や, モデルを構成する可能性のすべてが絶たれたので, S1 は充足不能であると結論できる.

MGTP では扱うモデルとして基底モデル[1] のみを考えるため, 入力節は次に述べる range-restrictedness の条件を満たしていることが必要である. 「後件部中に現れる変数は全て前件部に現れる」. この条件により前件部が基底論理式ならば後件部は基底論理式になる. 特に正節は基底論理式のみからなる. したがって, 生成されるモデルは全て基底原子論理式のみから構成されることになる.

Range-restrictedness の条件により, MGTP の扱える範囲は現実的には有限領域の問題になる. 一階の問題つまり, 無限領域を対象とする問題に対しては, 変数の基底代入例をエルブラン空間の全てに対して試みるので, 一般的に効率は悪くなる.

2.2 MGTP の特徴

MGTP の主な特徴は, 以下の 4 点である.

1. KL1 の言語特性を生かしたプログラミング：入力節集合が range-restrictedness の条件を満たしていればユニフィケーションが不要となる. この場合, 連言照合に KL1 の高速なガードユニフィケーションを用いることができる. また, 入力節集合は実行に先だって KL1 述語

[1]変数を含まないモデル

に変換される．これにより節探索に KL1 の節インデックスが利用でき，さらに新変数の生成も KL1 処理系に任すことができる．

2. 連言照合の冗長性の除去：M 個の要素からなるモデル候補に δ 個のモデルが追加された段階でリテラル数 2 個の前件部の連言照合を行おうとすると，

   $$(M+\delta)\times(M+\delta)=M\times M+M\times\delta+\delta\times M+\delta\times\delta$$

   通りの照合を行わなければならない．ここで $M\times M$ 通りの照合は既に行われているのでもし行うと冗長になる．この冗長を回避するために連言照合の履歴を保持する Ramified-Stack Algorithm[3] を取り入れている．

3. OR 並列性の抽出：節の後件部が選言(非ホーン節)の場合，そこでモデルの拡張に場合分けが必要となる．range-restrictedness の条件により，これらの拡張されたモデルは全て変数を含まない．これにより，モデル間の共有変数の問題がなくなり，それぞれ全く独立にモデルを拡張していくことができる．したがって非ホーン節でモデルを拡張する場合には，自然な OR- 並列実行が可能になる．

4. KL1 述語の呼び出し：前件部では KL1 ガード組み込み述語，後件部では KL1 のボディ組み込み述語とユーザ定義述語を呼び出すことができ，実用上便利である．

3 タブロ

タブロ法では論理式の分解を繰り返して，モデルを生成しようと試みる．モデル候補の中に F と $\neg F$ が含まれていれば(閉条件)，モデル生成に失敗したことになり，論理式の分解できなくなればモデル生成に成功したことになる．

3.1 命題タブロ

論理式をその形式に従って以下のように α 型と β 型に分ける．

α 型			β 型		
α	α_1	α_2	β	β_1	β_2
$F\wedge G$	F	G	$\neg(F\wedge G)$	$\neg F$	$\neg G$
$\neg(F\vee G)$	$\neg F$	$\neg G$	$F\vee G$	F	G
$\neg(F\supset G)$	F	$\neg G$	$F\supset G$	$\neg F$	G
$\neg\neg F$	F	F			

α 型 β 型の直感的解釈は次のようになる．

- 生成しようとしているモデルで α が成立するためには α_1 と α_2 が成立しなければならない．
- 生成しようとしているモデルで β が成立するためには β_1 または β_2 が成立しなければならない．

例として $((P\supset Q)\vee(P\supset R))\supset(P\supset(Q\vee R))$ の恒真性，つまり $\neg((P\supset Q)\vee(P\supset R))\supset(P\supset(Q\vee R))$ の充足不可能性の証明を示す．

1. $\neg((P\supset Q)\vee(P\supset R))\supset(P\supset(Q\vee R))$
2. $(P\supset Q)\vee(P\supset R)$　(1 に α)
3. $\neg(P\supset(Q\vee R))$　(1 に α)
4. P　(3 に α)
5. $\neg(Q\vee R)$　(3 に α)
6. $\neg Q$　(5 に α)
7. $\neg R$　(5 に α)
   (a) $P\supset Q$　(2 に β)
      i. $\neg P$　(7a に β)：閉(4)
      ii. Q　(7a に β)：閉(6)
   (b) $P\supset R$　(2 に β)
      i. $\neg P$　(7b に β)：閉(4)
      ii. R　(7b に β)：閉(7)

証明手続き $prove(ASetOfFormulas)$ の定義は大体表 1のようになる．証明すべき論理式が F のとき $prove(\{\neg F\})$ を実行し結果が $\top$ の場合に F の証明に成功したことになる．

3.2 様相命題タブロ

様相論理では通常の論理記号の他に $\Box$(必然性) と $\Diamond$(可能性) といった様相記号が取り扱われる [8]．$\Diamond$ は $\neg\Box\neg$ の略記である．様相タブロの理解を助けるために，まず簡単に Kripke モデルの解説をする．

3.2.1 Kripke モデル

このモデルは論理式の集合 (P)，可能世界の可算集合 (W)，到達可能関係と呼ばれる可能世界間の関係 $(R\subseteq W\times W)$，付値関数 $(m:P\rightarrow(W\rightarrow\{\top,\bot\}))$ を用いて定めることができる．$S=<W,R>$ を構造，$M=<W,R,m>$ をモデルと呼ぶ．世界 $w\in W$ における論理式 F の真偽値 $V_{M,w}(F)$ は次のように定義される．

1. F が命題変数の場合．$V_{M,w}(F):=m(F)(w)$
2. $F=\neg G$ の場合．
   $$V_{M,w}(\neg G):=\begin{cases}\top & V_{M,w}(G)=\bot\text{の時}\\ \bot & V_{M,w}(G)=\top\text{の時}\end{cases}$$
3. $F=F_1\wedge F_2$ の場合．
   $$V_{M,w}(F_1\wedge F_2):=\begin{cases}\top & \forall i\,V_{M,w}(F_i)=\top\\ \bot & \exists i\,V_{M,w}(F_i)=\bot\end{cases}$$

表 1: 命題タブロの証明手続き

$prove(Fs)$	:=	$\top$	$\exists F(F \in Fs \wedge \neg F \in Fs)$(閉条件)
		$prove(Fs \cup \{\alpha_1, \alpha_2\})$	$\exists F \in Fs(F$はα型$)$
		$prove(Fs \cup \{\beta_1\}) \wedge prove(Fs \cup \{\beta_2\})$	$\exists F \in Fs(F$はβ型$)$
		$\perp$	otherwise

4. $F = F_1 \vee F_2$ の場合.
$$V_{M,w}(F_1 \vee F_2) := \begin{cases} \top & \exists i\, V_{M,w}(F_i) = \top \\ \perp & \forall i\, V_{M,w}(F_i) = \perp \end{cases}$$

5. $F = F_1 \supset F_2$ の場合.
$$V_{M,w}(F_1 \supset F_2) := \begin{cases} \top & V_{M,w}(F_1) = \perp \text{または} V_{M,w}(F_2) = \top \\ \perp & \text{otherwise} \end{cases}$$

6. $F = \Box G$ の場合.
$$V_{M,w}(\Box G) := \begin{cases} \top & \forall w'(wRw' \supset V_{M,w'}(G) = \top) \\ \perp & \text{otherwise} \end{cases}$$

7. $F = \Diamond G$ の場合.
$$V_{M,w}(\Diamond G) := \begin{cases} \top & \exists w'(wRw' \wedge V_{M,w'}(G) = \top) \\ \perp & \text{otherwise} \end{cases}$$

Kripke の意味論では 世界 w で F が「必然的」であるのは w から到達できるいかなる世界においても F が真であると定義されるわけである.

3.2.2 様相タブロ

$\forall M \forall w\, V_{M,w}(F) = \top$ のとき F は恒真であると言う. ここで述べるタブロ法[1] はこの否定 $\exists M \exists w\, V_{M,w}(F) = \perp$ を仮定し, つまり, ある世界において $\neg F$ が成立するものと仮定し, 矛盾を導くことにより F の恒真性を証明する.

w を世界, F を論理式とするとき $w(F)$ を前置論理式という. 直感的には w において F が真であること意味する. 様相タブロは前置論理式を用いて定義される. 3.1 命題タブロと同様に前置論理式をその形式にしたがって α 型,β 型, γ 型, π 型に分ける (表 2).

γ 型, π 型 の直感的意味は大体以下のようになる.

- γ 型. $\Box F$ が世界 w おいて真であるとすると, w から到達できるいかなる世界 w' においても F は真である. $\Diamond F$ が w において偽であるとすると,w から到達できるいかなる w' においても F は偽である.
- π 型. $\Box F$ が w において偽であるとすると, F が偽であるような w' に w から到達できる. $\Diamond F$ が w において真であるとすると, F が真であるような w' に w から到達できる.

例として $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$ を体系 K において証明する.

1. $p(\neg(\Box(X \supset Y) \supset (\Box X \supset \Box Y)))$
2. $p(\Box(X \supset Y))$ (1 に α)
3. $p(\neg(\Box X \supset \Box Y))$ (1 に α)
4. $p(\Box X)$ (3 に α)
5. $p(\neg \Box Y)$ (3 に α)
6. $q(\neg Y)$ (5 に π. pRq)
7. $q(X)$ (4 に γ)
8. $q(X \supset Y)$ (2 に γ)
   (a) $q(\neg X)$ (8 に β) : 閉(7)
   (b) $q(Y)$ (8 に β) : 閉(6)

証明手続き $prove(ASetOfPrefizedFormula)$ も 様相タブロと同様に定義される (表 3). $\pi(w)$ が真なら w から到達可能なある w' において $\pi_1(w')$ が真である. w' の条件は w からの到達可能性だけなので上記のような条件が付く.

γ 型, π 型の分解は Predicate タブロの quantifier の分解に非常に似ている. 実際 π 型の w' の条件は eigen variable の条件と同じである. また, γ 型の分解において適切な w' の選択にユニフィケーションが有効であることも容易に想像できる.

4 MGTP 上の様相タブロ証明器

MGTP では空のモデルから始めモデルを矛盾が導かれるまで拡張していく. 拡張の仕方が複数ある場合はモデルを分岐拡張する. モデル生成という観点からみると入力節集合は, モデル生成の方法を指定していると見なすことができる. つまり, 正節 / 負節 / その他の節はそれぞれ, 初期モデル / モデル生成の失敗条件 / モデルの拡張規則を指定しているわけである.

これをタブロ法にあてはめると, 入力論理式は初期モデル, 分解規則はモデルの拡張規則, 閉条件はモデル生成の失敗条件と見なすことができる. MGTP ではモデルの要素は原始論理式のみであったが, これを一般の論理式に拡張する. このときタブロの (1)α 型, (2)β 型, (3)γ 型, (4)π 型 の分解規則は, M を生成しようとしているモデルとするとそれぞれ, (1)$\alpha \in M$ なら $\alpha_1 \in M$ かつ $\alpha_2 \in M$, (2)$\beta \in M$ なら

表 2: 様相命題タブロ

α 型			β 型			γ 型		π 型	
α	α_1	α_2	β	β_1	β_2	$\gamma(w)$	$\gamma_1(w')$	$\pi(w)$	$\pi_1(w')$
$w(F \wedge G)$	$w(F)$	$w(G)$	$w(\neg(F \wedge G))$	$w(\neg F)$	$w(\neg G)$	$w(\Box F)$	$w'(F)$	$w(\neg\Box F)$	$w'(\neg F)$
$w(\neg(F \vee G))$	$w(\neg F)$	$w(\neg G)$	$w(F \vee G)$	$w(F)$	$w(G)$	$w(\neg\Diamond F)$	$w'(\neg F)$	$w(\Diamond F)$	$w'(F)$
$w(\neg(F \supset G))$	$w(F)$	$w(\neg G)$	$w(F \supset G)$	$w(\neg F)$	$w(G)$				
$w(\neg\neg F)$	$w(F)$	$w(F)$							

表 3: 様相命題タブロの証明手続き

$prove(Fs)$	:=	$\top$	$\exists w \exists F(w(F) \in Fs \wedge w(\neg F) \in Fs)$(閉条件)
		$prove(Fs \cup \{\alpha_1, \alpha_2\})$	$\exists F \in Fs(F$はα型)
		$prove(Fs \cup \{\beta_1\}) \wedge prove(Fs \cup \{\beta_2\})$	$\exists F \in Fs(F$はβ型)
		$prove(Fs \cup \{\gamma_1(w')\})$	$\exists F \in Fs(F$はγ型)
		$prove(Fs \cup \{\pi_1(w')\})$	$\exists F \in Fs(F$はπ型. 但し w' は Fs に現れない)
		$\perp$	otherwise

$\beta_1 \in M$ または $\beta_2 \in M$, (3)$\gamma(w) \in M$ なら $\gamma_1(w') \in M$, (4)$\pi(w) \in M$ なら $\pi_1(w') \in M$ と自然に解釈できる.

4.1 問題と閉条件

問題 ($Theorem$) と閉条件は次のように表すことができる.

- 問題. $true \rightarrow p(\neg Theorem)$ (p はある定世界)
- 閉条件 $w(F), w(\neg F) \rightarrow false$

4.2 分解規則

分解規則は次のように表現することができる.

- α 型. $\alpha \rightarrow \alpha_1, \alpha_2$
- β 型. $\beta \rightarrow \beta_1; \beta_2$
- γ 型. $\gamma(w) \rightarrow \gamma_1(w')$
- π 型. $\pi(w) \rightarrow \pi_1(w')$

ここで,γ 型,π 型はこのままでは不完全であり w' を具体的に与える必要がある. この与え方は到達可能関係 R が満たす性質により異なる. R の性質により様々な様相が考えられる. 以下, R の性質ごとに γ 型,π 型を考察していく.

1. 無条件 -K-. R に何の条件もない場合の様相を K という. この場合が最も簡潔である.
   - γ 型. $\gamma(w), wRw' \rightarrow \gamma_1(w')$
   - π 型. $\pi(w) \rightarrow \{\{\text{new\_world}(w')\}\}, wRw', \pi_1(w')$

   $\{\{...\}\}$ は KL1 述語呼び出しを表す. π 型の w' の条件を満たすために KL1 述語が呼ばれる.

2. 反射律 -T-. R が反射律を満たす場合, それぞれの世界からそれ自身へ到達可能であるから, K に以下のものを加える.
   - γ 型. $\gamma(w) \rightarrow \gamma_1(w)$

3. 推移律 -$K4$-. R が推移律を満たすので, 推移律を表す規則を K に加える.
   - 推移律. $wRw', w'Rw'' \rightarrow wRw''$

4. 反射律と推移律 -$S4$-. T と $K4$ の両性質を持つので K に次の規則を加える.
   - γ 型. $\gamma(w) \rightarrow \gamma_1(w)$
   - 推移律. $wRw', w'Rw'' \rightarrow wRw''$

5. 同値関係 -$S5$-. R が同値関係 (反射律, 対称律, 推移律) を満たす時, この様相を $S5$ という. $S5$ においては全ての世界が互いに到達可能と考えられるから, 証明の過程で現れるどの世界も互いに到達可能となる. そこで, $world(World)$ という特別な述語を使い, 証明の過程で現れる世界を全てモデルに登録することにする.
   - 問題. $true \rightarrow p(\neg Theorem), world(p)$
   - γ 型. $\gamma(w), world(w') \rightarrow \gamma_1(w')$
   - π 型. $\pi(w) \rightarrow \{\{\text{new\_world}(w')\}\}, world(w'), \pi_1(w')$

4.3 問題点

4.3.1 停止性

上記手続きは推移律を含まない体系 (K や T など) では停止するが, 推移律を満たすような体系 ($K4$ や $S4$ など) で

は上記手続きは停止しない場合がある [1]. 例えば $\neg(\Diamond F \wedge \Box\Diamond F)$ を $K4$ において証明を試みよう.

1. $p(\Diamond F \wedge \Box\Diamond F)$
2. $p(\Diamond F)$ (α)
3. $p(\Box\Diamond F)$ (α)
4. $p_1(F)$ (pRp_1) (π)
5. $p_1(\Diamond F)$ (γ)
6. $p_2(F)$ (p_1Rp_2) (π)
7. $p_2(\Diamond F)$ (γ)
8. $\vdots$

容易に想像できるように $p_1, p_2, \ldots$ と永遠に世界が続き, $\forall i(p_i(F) \wedge p_i(\Diamond F))$ となる.

$F_p = \{F \mid p(F)\}$ とすると, 一般に (1) $\exists p \exists q(pRq \wedge F_p \supset F_q)$ が成り立つ時, 世界の系列 $p, \ldots, q$ からモデルが構成できる. つまり, この世界の系列からは矛盾は出てこない. したがって, (1) がある世界の系列において成り立つなら, その系列を拡張する必要はない. また, 手続きが停止しないのはこのような場合のみである.

系列を拡張するのは π 型の場合であるので, (1) を確かめてから π 型を適用するようにすれば, 世界の無限系列を作ることはない. したがって, 手続きの停止性を保証することができる. しかしこの確認の手間は高価なものである.

実用的な他の方法は, 適当な制限値を設けて $p_1, p_2, \ldots$ の拡張をある長さに制限してしまうことである. 各世界に最初の世界[2] からの距離を添付しておき, これが制限値を越えた場合には π 型を適用しないようにすれば良い. これは容易に埋め込むことができる.

- 問題. $true \rightarrow p_N(\neg Theorem)$ (N が制限値)
- π 型.

$$\pi(w_N), \{\{N \geq 0\}\} \rightarrow \{\{\mathrm{new\_world}(w')\}\}, w_N R w'_{N-1}, \pi_1(w'_{N-1})$$

この方法を採用する場合に考慮しなければならないのは, 必要にして十分な制限値の設定である. 制限値が小さ過ぎると証明ができるかも知れないのに証明できないし, 逆に制限値が大き過ぎると無駄な世界の拡張を行うからである. 十分に大き過ぎる制限値としては 2^n (n は $Theorem$ に含まれる部分論理式の個数[3]) があるがこれは現実的ではない. また, これを少し改良して $2^{n'}$ (n' は $Theorem$ に含まれる部分論理式で一番外側の論理記号が様相記号であるものの個数) であることが分かるが, 本質的な改良ではない.

もう少し実用的な制限値の設定は今後の課題である.

4.3.2 優先度

タブロ法による自動証明では分解規則の適用順序が効率上重要な要因となる. 実際 様相命題タブロでは $\alpha, \beta, \gamma, \pi$ 順の優先度がかなり一般的な Heuristics である [2].

現在の MGTP では入力節間の優先度を指定することができないので, このような Heuristics を確かめてみることができない. この点については MGTP の機能拡張により対処可能である.

5 記述例

MGTP での K と PTL (Propostional Temporal Logic) の記述例を掲載する.

5.1 K

$p(F)$ を KL1 の 2 要素ベクタ {F,p} と表現することにする. また論理式 $\neg F$, $F \wedge G$, $F \vee G$, $F \supset G$, $\Box F$, $\Diamond F$, pRq はそれぞれ KL1 では -F, F*G, F+G, F>G, box(F), dia(F), path(p,q) と表すことにする. また, new_atom(A) は新たなアトムを生成する KL1 の組み込みである.

```
% Close Condition
{-F,P},{F,P} --> false.
% Not
{-(-F),P} --> {F,P}.
% Or
{F+G,P} --> {F,P};{G,P}.
{-(F+G),P} --> {-F,P},{-G,P}.
% And
{F*G,P} --> {F,P},{G,P}.
{-(F*G),P} --> {-F,P};{-G,P}.
% Implication
{F>G,P} --> {-F,P};{G,P}.
{-(F>G),P} --> {F,P},{-G,P}.
% Box
{box(F),P},path(P,Q) --> {F,Q}.
{-box(F),P} --> {{new_atom(Q)}},path(P,Q),{-F,Q}.
% Diamond
{dia(F),P} --> {{new_atom(Q)}},path(P,Q),{F,Q}.
{-dia(F),P},path(P,Q) --> {-F,Q}.
```

5.2 PTL

PTL では次の時刻に世界がとりうる状態は唯一に決まる, 線形で離散的な時間構造に基づいている. 様相記号には $\Box$, $\Diamond$ の他に $\bigcirc$(next) と U(Until) がある. 以下 Kripke 流に意味を記述する.

[2] $true \rightarrow p(\neg Theorem)$ における世界 p

[3] 部分論理式の性質より F_p の可能性として $Theorem$ に含まれる部分論理式の集合のみを考えれば良いから

5.2.1 PTL の Kripke 流モデル

可能世界の集合 T は非負整数の集合 $\{0,1,2,\ldots\}$ で通常の $=$ と $<$ が定義されているものとする. 0 は現時点と解釈される. $V_{M,t}(F)$ で時刻 $t \in T$ における論理式 F の真偽値を表すものとする.

1. F が命題変数の場合. $V_{M,t}(F) := m(F)(t)$
2. $F = \neg G$ の場合. $V_{M,t}(\neg G) := \begin{cases} \top & V_{M,t}(G) = \bot \text{の時} \\ \bot & V_{M,t}(G) = \top \text{の時} \end{cases}$
3. $F = F_1 \wedge F_2$ の場合.
$$V_{M,t}(F_1 \wedge F_2) := \begin{cases} \top & \forall i\, V_{M,t}(F_i) = \top \\ \bot & \exists i\, V_{M,t}(F_i) = \bot \end{cases}$$
4. $F = F_1 \vee F_2$ の場合.
$$V_{M,t}(F_1 \vee F_2) := \begin{cases} \top & \exists i\, V_{M,t}(F_i) = \top \\ \bot & \forall i\, V_{M,t}(F_i) = \bot \end{cases}$$
5. $F = F_1 \supset F_2$ の場合.
$$V_{M,t}(F_1 \supset F_2) := \begin{cases} \top & V_{M,t}(F_1) = \bot \text{または} V_{M,t}(F_2) = \top \\ \bot & \text{otherwise} \end{cases}$$
6. $F = \Box G$ の場合.
$$V_{M,t}(\Box G) := \begin{cases} \top & \forall t'(t \le t' \supset V_{M,t'}(G) = \top) \\ \bot & \text{otherwise} \end{cases}$$
7. $F = \Diamond G$ の場合.
$$V_{M,t}(\Diamond G) := \begin{cases} \top & \exists t'(t \le t' \wedge V_{M,t'}(G) = \top) \\ \bot & \text{otherwise} \end{cases}$$
8. $F = \bigcirc G$ の場合.
$$V_{M,t}(\bigcirc G) := \begin{cases} \top & V_{M,t+1}(G) = \top \\ \bot & \text{otherwise} \end{cases}$$
9. $F = F_1 \mathrm{U} F_2$ の場合.
$$V_{M,t}(F_1 \mathrm{U} F_2) := \begin{cases} \top & \exists t'(t \le t' \wedge V_{M,t'}(F_2) = \top \wedge \forall t''(t \le t'' < t' \supset V_{M,t''}(F_1 \wedge \neg F_2) = \top)) \\ \bot & \text{otherwise} \end{cases}$$

$\forall M V_{M,0}(F) = \top$ の時 F は恒真であるという. したがって証明はこの否定 $\exists M V_{M,0}(\neg F) = \top$ を仮定し矛盾を導くことになる.

5.2.2 MGTP 入力節

PTL には $\bigcirc$ と U の様相記号があるので, s(T) (時刻 T の次の時刻) と first(T,F) (時刻 T 以後論理式 F が成立する の最初の時刻) の二つの記法を導入する.

$\bigcirc$ と U に関する分解規則は次のようになる.

$$t(\bigcirc F) \to s(t)(F), t < s(t)$$

$$t(\neg \bigcirc F) \to t(\bigcirc \neg F)$$

$$t(F_1 \mathrm{U} F_2) \to$$
$$first(t, F_2)(F_2), t \le first(t, F_2), \mathrm{u1}(t, first(t, F_2), F_1)$$

$$t(\neg(F_1 \mathrm{U} F_2)) \to t(\Box(\neg F_2)); \mathrm{u2}(F_1, F_2, t)$$

ここで, $s(t)$ は t の次の時間, $\mathrm{u1}(t, t_1, F)$ は $\forall t'(t \le t' < t_1 \supset t'(F))$ を $u2(F_1, F_2, t)$ は $\exists t_1 > t(t_1 = first(t, F_2) \wedge \exists t'(t \le t' < t_1 \wedge t'(\neg F_1)))$ を表している.

$\bigcirc F$ を cir(F), FUG を F \/ G と表記することにして, PTL の規則を表現すると次のようになる. 各論理記号に関する規則の他に時刻の線形性や同値性に関する規則が必要となる. また時刻 t を {T,L,Lb} と表現し T は t に対する時刻, L は時間の正方向の拡張の制限値, Lb は時間の負方向の拡張の制限値を表している. 負方向の拡張というのは, $\bigcirc$ に関する証明のために, 時刻 t に対してその直前の時刻 $p(t)$ を考慮する必要があるため, 導入した. 時刻によらず $L + Lb$ は一定であるように記述されている.

```
% Close Condition
{-F,T},{F,T} --> false.
{-F,T},{F,T1}, T=T1 --> false.
% Not
{-(-F),T} --> {F,T}.
% Or
{F+G,T} --> {F,T};{G,T}.
{-(F+G),T} --> {-F,T},{-G,T}.
% And
{F*G,T} --> {F,T},{G,T}.
{-(F*G),T} --> {-F,T};{-G,T}.
% Implication
{F>G,T} --> {-F,T};{G,T}.
{-(F>G),T} --> {F,T},{-G,T}.
% equivalence
{F=:=G,T} --> {F*G,T};{-(F+G),T}.
{-(F=:=G),T} --> {-(F>G),T};{-(G>F),T}.
% Box
{box(F),T}, T<U --> {F,U}.
{box(F),T}, T=U --> {F,U}.
{box(F),T} --> {F,T}.
{-box(F),T} --> {dia(-F),T}.
% Diamond
{dia(F),T}, {{ T= {T0,L,Lb}, L>0}} -->
    {F,U}, T=<U, {{U = {first(T0,F),~(L-1),~(Lb+1)}}}.
{-dia(F),T} --> {box(-F),T}.
% Next
{cir(F),T}, {{ T= {T0,L,Lb}, L>0}} -->
    {{pred(T0, YN)}}, cir, cir1(YN, F,T).
cir1(yes, F,T), {{T = {p(T0),L,Lb}}} -->
                                      % T<ST : already in Model
    {F,ST}, {{ST = {T0,~(L-1),~(Lb+1)}}}.
cir1(no, F,T), {{T = {T0,L,Lb}}} -->
    {F,ST}, T<ST, {{ST = {s(T0),~(L-1),~(Lb+1)}}}.
```

```
{-cir(F),T} --> {cir(-F),T}.
cir, T<U, {{U={U0,M,Mb}, Mb>0}} -->
    {{succ(U0, YN)}}, cir2(YN, T,U).
cir2(no, T,{U0,M,Mb}) -->
    T=<U1, {{U1={p(U0),~(M+1),~(Mb-1)}}}.
% successor and predecessor
{T0,_,_} < U, U < {s(T0),_,_} --> false.
{p(T0),_,_} < U, U < {T0,_,_} --> false.
% Until
{F\/G,T}, {{ T={T0,L,Lb},L>0}} -->
    {G,U}, T=<U, '\/'(T,U,F),
    {{U = {first(T0,G),~(L-1),~(Lb+1)}}}.
'\/'(T,U,F),T<U1,U1<U --> {F,U1}.
'\/'(T,U,F),T=U1,U1<U --> {F,U1}.
'\/'(T,U,F),T<U --> {F,T}.
{-(F\/G),T}, {{ T={T0,L,_},L>0}} -->
    {box(-G),T};until(F,G,T).
until(F,G,T),{{ T={T0,L,Lb}}} -->
    {G,U1}, {NF,U2}, T=<U2, U2<U1,
    {{negate(F,NF),L1 := L-1,Lb1 := Lb+1,
    U1 = {first(T0,G),L1,Lb1},
    U2 = {first(T0,NF),L1,Lb1}}}.
% first
{T0,_,_}<U, U<{first(T0,F),_,_} -->
    {{negate(F,NF)}}, {NF,U}.
{T0,_,_}=U, U<{first(T0,F),_,_} -->
    {{negate(F,NF)}}, {NF,U}.
T<{first(T0,F),_,_}, {{T={T0,_,_}}} -->
    {{negate(F,NF)}}, {NF,T}.
% equality and inequality
T=<U --> T=U;T<U.
%T<T1,T1<T2,{{T\=T2}} --> T<T2.
T<T1,T1<T2 --> {{equal(T,T2, YN)}}, '<'(YN, T,T2).
'<'(yes, _,_) --> false.
'<'(no, T,T2) --> T<T2.
%T<T1,T<T2,{{T1\=T2}} --> T1<T2;T1=T2;T2<T1.
T<T1,T<T2 --> {{equal(T1,T2,YN)}}, '<1'(YN,T1,T2).
'<1'(no,T1,T2) --> T1<T2;T1=T2;T2<T1.
%T1<T,T2<T,{{T1\=T2}} --> T1<T2;T1=T2;T2<T1.
T1<T,T2<T --> {{equal(T1,T2,YN)}}, '<1'(YN,T1,T2).
%T=T1,T1<T2,{{T\=T2}} --> T<T2.
T=T1,T1<T2 --> {{equal(T,T2,YN)}}, '<'(YN, T,T2).
%T<T1,T1=T2,{{T\=T2}} --> T<T2.
T<T1,T1=T2 --> {{equal(T,T2,YN)}}, '<'(YN, T,T2).
%T=T1,T1=T2,{{T\=T2}} --> T=T2.
T=T1,T1=T2  --> {{equal(T,T2,YN)}}, '='(YN, T,T2).
'='(no,T,T2) --> T=T2.
% successor and predecessor
%s(T)<U --> T<p(U).
%s(T)=U --> T=p(U).
T<U,{{ T={s(T0),_,_}, U={U0,_,_}}} -->
    {{succ(U0, YN)}}, '<2'(YN, T,U).
'<2'(yes, {s(T0),L,Lb},{s(U0),M,Mb}) -->
    T1<U1, {{T1={T0,~(L+1),~(Lb-1)},
    U1={U0,~(M+1),~(Mb-1)}}}.
'<2'(no, T,U), {{T={s(T0),L,Lb}, U={U0,M,Mb}, Mb>0}} -->
    T1<U1,U1<U,
    {{T1={T0,~(L+1),~(Lb-1)}, U1={p(U0),~(M+1),~(Mb-1)}}}.
T=U,{{T={s(T0),_,_}, U={U0,_,_}}} -->
    {{succ(U0, YN)}}, '=2'(YN, T,U).
'=2'(yes, {s(T0),L,Lb},{s(U0),M,Mb}) -->
    T1=U1,
    {{T1={T0,~(L+1),~(Lb-1)}, U1={U0,~(M+1),~(Mb-1)}}}.
'=2'(no, T,U), {{T={s(T0),L,Lb}, U={U0,M,Mb}, Mb>0}} -->
    T1=U1,U1<U,
    {{T1={T0,~(L+1),~(Lb-1)}, U1={p(U0),~(M+1),~(Mb-1)}}}.
% reflective
T=T1 --> T1=T.
% KL1 Predicates
equal(T,T, YN) :- YN = yes. otherwise.
equal(_,_, YN) :- YN = no.

succ(s(X), YN) :- YN = yes. otherwise.
succ(X, YN) :- YN = no.

pred(p(X), YN) :- YN = yes. otherwise.
pred(X, YN) :- YN = no.

negate(-F,NF) :- NF = F. otherwise.
negate(F,NF) :- NF = -F.
```

6 計測

PTL の定理 39 題[4] を PSI-II 上の擬似マルチ PSI システムで解いた時の実行時間を記載する (表 4). 参考までに[4] による証明時間も併記する (ALS の欄). これは置換規則を用いて証明された時間である.

6.1 戦略

計測には MGTP の三種類の推論戦略を用いた. それぞれの戦略は以下のとおり.

- 負節優先. モデル生成のあるステージで生成された後件部の全てを用いて, モデルの拡張を負節 (閉条件) の前件部照合をしながら行う. 負節の前件部照合に失敗したモデルに対して次の生成ステージに進む.
- Horn 節優先. 後件部が単言である節によるモデル拡張を, 選言である節による拡張より優先的に行なう. 拡張の仕方を二種類与えた.
  - 漸進. モデルへの要素の追加を一つ行っては, 連言照合を行う.
  - 急進. モデルへの要素の追加を行えるだけ行ってから, 連言照合を行う.

時刻拡張の制限値は入力論理式を木表現したときの根から葉までの路での様相記号の数の最大値とした. たとえば, $\Diamond\Box A \supset \Box\Diamond\Box A$ の場合の制限値は 3 となる.

表 4: 命題時相論理の証明時間

番号	問題	負節優先 (red/msec)	漸進 (red/msec)	急進 (red/msec)	ALS (msec)
1	$\Diamond\Box a \supset \Box\Diamond\Box a$	**186862/9297**	**187236/78963**	**1579449/80704**	6383
2	$\neg\Diamond a \equiv \Box\neg a$	5560/267	8254/326	5169/258	6617
3	$\Diamond a \vee \Diamond\neg a$	2127/158	3179/179	1922/156	683
4	$a \supset \Diamond a$	1750/143	2058/143	1677/142	500
5	$\bigcirc a \supset \Diamond a$	2414/170	6947/312	3199/214	1783
6	$\Box a \equiv \Box\Box a$	11772/493	17320/655	14723/598	3050
7	$\Diamond a \equiv \Diamond\Diamond a$	10394/458	16464/609	13044/553	3367
8	$\Diamond\neg a \equiv \neg\Box a$	5951/283	9235/370	5749/278	1583
9	$\Box(a \supset b) \supset (\Diamond a \supset \Diamond b)$	7059/361	13039/509	8335/409	3067
10	$\Box(a \wedge b) \equiv (\Box a) \wedge (\Box b)$	19951/838	28023/980	19392/780	10250
11	$\Diamond(a \vee b) \equiv (\Diamond a) \vee (\Diamond b)$	20505/900	28108/1009	19306/798	9717
12	$(\Box a \vee \Box b) \supset \Box(a \vee b)$	9437/434	15496/580	12252/499	3250
13	$\Diamond(a \wedge b) \supset (\Diamond a \wedge \Diamond b)$	9035/444	12106/458	15635/622	3317
14	$(\Box a \wedge \Diamond b) \supset \Diamond(a \wedge b)$	6493/343	9359/383	7947/412	2967
15	$\bigcirc(a \wedge b) \equiv (\bigcirc a \wedge \bigcirc b)$	7817/352	21064/700	9580/457	8283
16	$\bigcirc(a \vee b) \equiv (\bigcirc a \vee \bigcirc b)$	8476/376	18777/624	10645/500	8317
17	$\bigcirc(a \supset b) \equiv (\bigcirc a \supset \bigcirc b)$	8492/373	17975/590	10289/482	8367
18	$\bigcirc(a \equiv b) \equiv (\bigcirc a \equiv \bigcirc b)$	10435/447	28664/938	13280/567	19033
19	$\bigcirc\Box a \equiv \Box\bigcirc a$	6290/307	219017/8120	49143/2270	8800
20	$\bigcirc\Diamond a \equiv \Diamond\bigcirc a$	9080/415	230890/8819	35718/1722	9000
21	$\Box\Diamond\Box a \equiv \Diamond\Box a$	192859/9420	**1889170/79442**	**1593910/81491**	46983
22	$\Diamond\Box\Diamond a \equiv \Box\Diamond a$	108212/5193	**3063935/127256**	698682/35792	44467
23	$\Box a \equiv (a \wedge \bigcirc\Box a)$	7985/365	120439/4799	62770/2732	30867
24	$\Diamond a \equiv (a \vee \bigcirc\Diamond a)$	9564/425	37586/1342	18953/869	27167
25	$(a \wedge \Diamond\neg a) \supset \Diamond(a \wedge \bigcirc\neg a)$	8079/396	27135/1001	40002/1827	117167
26	$(\neg a)\mathbf{U}a \equiv \Diamond a$	11082/488	14051/514	14003/602	6700
27	$\Box a \wedge \Diamond b \supset a\mathbf{U}b$	14424/687	19294/664	25346/1102	6350
28	$(a\mathbf{U}b)\mathbf{U}b \supset a\mathbf{U}b$	18080/874	22676/820	13095/596	14833
29	$a\mathbf{U}(a\mathbf{U}b) \supset a\mathbf{U}b$	191819/9438	116494/4441	92335/4579	25133
30	$\Box a \wedge (b\mathbf{U}c) \supset (a \wedge b)\mathbf{U}(a \wedge c)$	81886/3966	74848/2777	86968/4364	34983
31	$(a \wedge b)\mathbf{U}c \equiv (a\mathbf{U}c) \wedge (b\mathbf{U}c)$	87486/4114	95553/3533	87232/3963	27033
32	$a\mathbf{U}(b \vee c) \equiv (a\mathbf{U}b) \vee (a\mathbf{U}c)$	741784/35072	884795/34496	391959/18794	236183
33	$(\Diamond a \vee \Diamond b) \supset (((\neg a)\mathbf{U}b) \vee ((\neg b)\mathbf{U}a))$	50996/2297	5488/1976	31987/1391	119367
34	$a\mathbf{U}(b \wedge c) \supset (a\mathbf{U}b) \wedge (a\mathbf{U}c)$	121420/5657	117613/4504	135576/6514	32417
35	$(a\mathbf{U}c) \vee (b\mathbf{U}c) \supset (a \vee b)\mathbf{U}c$	35914/1646	49872/1755	32479/1450	11817
36	$(a \supset b)\mathbf{U}c \supset (a\mathbf{U}c \supset b\mathbf{U}c)$	31036/1466	35558/1301	30501/1469	10000
37	$(a\mathbf{U}b) \wedge ((\neg b)\mathbf{U}c) \supset a\mathbf{U}c$	158058/7994	62981/2324	99481/4714	33583
38	$(a\mathbf{U}b)\mathbf{U}c \supset (a \vee b)\mathbf{U}c$	74924/3663	56965/2087	188234/9058	20267
39	$a\mathbf{U}(b\mathbf{U}c) \supset (a \vee b)\mathbf{U}c$	**3443127/165754**	**2164955/87924**	**1567969/79393**	33367

6.2 考察

MGTP の戦略間,MGTP と [4] 間の二点に分けて考察する. [4] のシステムは ALS-Prolog 上に実現されいるが,性能評価に用いた計算機の性能及び置換規則の詳細が不明なため,厳密には比較できない.

1. 負節優先と Horn 節優先：全般的に負節優先が優れている. 一般的には Horn 節優先は,場合分けが先送りされ効率が良さそうである. しかし場合分けが本質的な場合は,場合分けを先につまり,非ホーン節の適用を早めに行なった方が,速く証明が見つかるであろう. PTL の例題ではそのような場合が,ほとんどであったと思われる. また,漸進型と急進型において違いが顕著なもの[4]がある. これは,証明図の作り方の微妙な違いが,探索空間の大きさに影響を与えたためであると思われる.

2. MGTP と置換規則： MGTP の入力節は,前件部のパターンを後件部のパターンで置き換える,と解釈することもできる. 置換規則 [4] というのは,そのように解釈した場合の節集合であり, range-restrictedness の条件がないので,置換規則による証明は MGTP による証明より範囲が,従って探索空間が広い. MGTP がほとんど場合において優れているのは,この探索空間の大小によるもとのと思われる. つまり,置換規則による証明では無駄な探索空間の枝に入り込む危険性が MGTP に比べて大きく,それが証明時間に現れているもの思われる. もちろん危険を侵したことにより,早く探索に成功する場合もあり,置換規則による証明が優れている問題は,そのような状況に遭遇したものと想像される.

7 おわりに

タブロ法の分解規則を MGTP の入力節として表現することにより, MGTP 上にタブロ証明器を作成した. タブロ法の分解規則と閉条件に対応する記述しかないという点でこの証明器は非常に簡潔である. 入力節をプログラム言語とみなすことにすると 12 行で様相命題論理 K の証明器が作れたことになる. これはタブロ証明器製作者にとって非常に有益である. 各種様相タブロ証明器を比較的容易に作成することができ,思考実験が気軽にできるようになる.

定理証明器を作成するにあたって,

(1) 証明対象領域の性質の表現法

(2) 推論速度の高速化

は重要な要因となる. (1) は理学的,(2) は工学的側面をもつ. いくら対象領域を巧みに表現したとしても推論速度が遅くては実用的ではないし,その逆も言える.

[4] 19,20,22 など

本論文で述べた二段階証明器作成法では,(1) は MGTP の入力節の表現に, (2) は MGTP 自体の高速化に帰着される. 二つの要因を分離することで,見通しのよい証明器の作成が可能となった. もちろん (1) のレベルより (2) のレベルで対処したほうが現実的なものもある. 等号に関するものや AC-単化などはその例であろう. このようなことはソフトウェアとハードウェアの関係にも似ていて興味深い.

残されている大きな問題として停止性 (4.3.1参照) がある. 世界拡張回数の上限値を理論的上限値に設定すれば理論的停止性はいえる. 理論的上限値の代わりに現実的上限値を見つけるは今後の課題である. もう一つの現実的アプローチとして Matrix 法の証明器 [7] を MGTP 上に作成する方法がある. この場合,タブロ法も Matrix 法も Sequent Calculus を基礎としているため,本論で述べた手法は生かせると思われる.

謝辞

様相論理に関して多くの助言をいただいた津田塾大学数学科の磯田恵以子さんに感謝します. また, ICOT 自動証明グループの方々に感謝します.

参考文献

[1] Fitting, M.: *Proof Methods for Modal and Intuitionistic Logic*, D.Reidel Publishing Co., Dordrecht 1983.

[2] Fitting, M.: "First-Order Modal Tableaux", *Journal of Automated Reasoning*, Vol.4, No.2, 1988.

[3] Fujita, H., Hasegawa, R.: "A Model Generation Theorem Prover Using A Ramified-Stack Algorithm" *ICOT TR-606*, 1990.

[4] D.A. Plaisted and S-J. Lee : "Inference by Clause Linking", 1990.

[5] R.M. Smullyan : *First-Oder Logic*, Vol 43 of Ergebnisse der Mathematik, Springer-Verlag, Berlin, 1968.

[6] Ueda, K., Chikayama, T.: "Design of the Kernel Language for the Parallel Inference Machine" , Computer J., Dec. 1990.

[7] L.A. Wallen : *Automated Proof Search in Non-Classical Logics*, MIT Press, 1990.

[8] 米崎 直樹 : 様相論理, 情報処理, Vol.30, No.6, 1989.