遺伝的プログラミングを用いた命題 ME タブロー法による定理自動証明

A Propositional Model Elimination Tableau Based on Genetic Programming

岩沼　宏治* 　矢ヶ崎　剛史*,†
Koji Iwanuma 　Takeshi Yagasaki

* 山梨大学工学部コンピュータ・メディア工学科
Dept. of Computer Science and Media Engineering, Yamanashi University, Kofu 400-8511, Japan.

1998年6月15日　受理

Keywords: genetic programming, automated deduction, theorem proving, tableau, model elimination.

Summary

In this paper, we study automated theorem proving based on Model Elimination (ME) tableau and genetic programming. We apply several GP-based search methods to propositional ME tableau calculus, and evaluate the performance of them experimentally.

1. はじめに

定理自動証明問題において，その巨大な探索空間の効率的な探索法の開発は重要な研究課題である．代表的な探索戦略として OTTER の Ratio3 [McCune 92] がある．Ratio3 は重み関数を用いた局所的探索と幅優先探索による大域的探索を併用・統合した戦略で，大きな成功をおさめている．

局所的探索と大域的探索のより良い統合手法の開発は重要である．遺伝的アルゴリズム [伊庭 96, 北野 97] はその枠組として有望と考えられるが，現在までのところ殆んど研究されていない．筆者の知る限り，コンビネータ論理上の存在変数の推定問題 [Fuchs 97] や，Horn 問題に対する MGTP のモデル候補推定問題 [中山 98] に関する研究しかない．そこで本研究では，遺伝的プログラミング（Genetic Programming, 以下 GP）技術を用いた ME タブロー型の定理自動証明について考察を行なう．

本論文でのタブローとは「場合分けの木」を意味し，各節点にリテラルがラベル付けされた木である．タブローに基づく定理自動証明とは，閉じたタブロー木，即ち全ての場合について矛盾が検出された木の導出計算である．よってプログラムなどの木構造データの確率的な探索を行なう GP 手法と，タブロー型定理証明は基本的に親和性がある．

本論文ではモデル消去法（Model Elimination, 以下 ME）[Loveland 69] に基づく ME タブローを扱う．ME および ME タブローには多く研究 [Baumgartner 98, Letz 94, Moser 97] があり，GP を基礎とする定理証明でもこれらの既存の結果が援用できる．

我々の最終目標は一階論理での定理自動証明である．本論文ではその準備として，命題論理での GP を用いた ME タブロー型定理証明について考察を行なった．幾つかの技法を考察し，比較的良い実験的評価が得られたので報告する．なお以下に述べる証明技法は命題論理の高速な定理証明を目的としたものではない．あくまで一階論理への拡張を念頭におき，その基礎として開発した一般的な技法であることを予めお断りしておく．

2. ME タブロー

ME は順序付線形導出法の一つであり，数多くの定理証明プログラムが実際に開発されてきている [Astrachan 94, Iwanuma 97, Iwanuma 98a, Iwanuma 98b, Moser 97, Stickel 92]．近年 ME はタブローを用いて定式化されることが多い [Baumgartner 98, Letz 94].

【定義 1】（タブロー，ME タブロー） S を命題論理上の節集合とする．S上のタブロー Tとは以下の条件を満たす順序対 $\langle t, \lambda \rangle$ である．

(1) t は順序木.

(2) λは t の根以外の節点へリテラルをラベル付けする関数で以下の条件を満たすもの．

- 親が同じ（即ち兄弟関係の）節点 $N_1,\dots,N_k$ のラベル集合 $\{\lambda(N_1),\dots, \lambda(N_k)\}$ は Sの節である．

タブロー Tの頂上節とは，根の直接の子節点 $M_1,\dots,M_k$ のラベルの集合 $\{\lambda(M_1),\dots, \lambda(M_k)\}$ からなる Sの節を言う．また Tの大きさ $|T|$ を Tの節点数と定める．

Tが ME タブローであるとは，根以外の任意の内部節

† 現在，沖ソフトウエア株式会社勤務.

点Nに対して，ラベルが相補的（即ち$\lambda(N) = \neg\lambda(N')$）な直接の子節点$N'$が存在する場合を言う．

【定義 2】（道，閉道，開ゴール） タブローTの道b（branch）とは，根から葉へ辿った節点の列$N_1, \ldots, N_n$である．bが閉道であるとは，bのラベル$\lambda(N_1), \ldots, \lambda(N_n)$の中に相補的なリテラルの対が存在する場合を言う．閉道でない道を開道と呼ぶ．bの長さ$\mathrm{D}(b)$はb中の節点の数nと定める．bの葉N_nを$\mathrm{leaf}(b)$と表記し，bが閉じていない場合，$\mathrm{leaf}(b)$を開ゴールと呼ぶ．

【定義 3】（閉タブロー） タブローTが閉じているとは，T中の全ての道が閉じている場合を言う．

［定理 1］（MEタブローの完全性） Sを命題節集合とする．Sが充足不能であることと，S上の閉じたMEタブローが存在することは同値である．

証明は [Letz 94] を参照して頂きたい．

〔例 1〕 Sを以下の充足不可能な節集合とする．

(1) $\neg p$　　(2) $p \vee \neg q \vee \neg r$

(3) $p \vee q$　　(4) r

図1はS上の閉じたMEタブローの例である．図中の$*$は閉道を，点線は相補的なリテラルの対を表している．

3. GPを用いたMEタブロー型の定理証明

GPは多点情報を利用する確率的探索・学習手法の一つであり，プログラムに代表される木のような構造データの生成を目的としている．基本的には (1) 初期集団を生成し，終了条件が満足されるまで (2-a) 適応度の評価・選択，(2-b) 交叉，(2-c) 突然変異を順次繰り返し適用していく．

GPを適用するにあたって，一般に以下のような点を考慮する必要性が指摘されている [伊庭 96]．

(1) 遺伝子の木構造は良好な形質が次世代に確実に伝わり，進化を促進するような表現が望ましい．

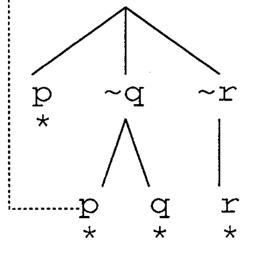


図1 閉じたMEタブロー

(2) 盲目的な交叉・突然変異は有益な部分構造（スキーマ）を破壊しやすい．木構造の意味を考慮する必要がある．

(3) 遺伝子の適応度の評価も，木の意味を十分に反映する必要がある．

(4) 冗長な探索を避け，より有益で大きな遺伝子を合成するために，適切な部分構造を保存し再利用する必要がある．

以上を考慮して，次のような遺伝的操作を考える．

3・1 遺伝子構造

本研究ではGPによる閉タブローの探索を試みる．遺伝子としてはMEタブローそのものを考える．S上のMEタブローの開ゴールを$L_1, \cdots, L_n$とするとき，論理和$L_1 \vee \cdots \vee L_n$はSの論理的帰結をなし，仮説推論と極めて密接な関係がある [Inoue 92]．GPによるMEタブロー木の探索は，ある種の確率的な仮説推論と見なすこともでき，極めて自然な枠組である．

次世代に良好な形質を遺伝させる一つの方策として，個々のタブローを小さくすることが考えられる．与えられた節集合の個々の節が長い場合，MEタブロー木は横方向に長くなる．部分的に有用な構造を含んでいても，タブロー全体の適応度を上げることは難しく，次世代まで生き残ることは難しい．GPの枠組では，小さく有用な部分構造・スキーマを次世代に残し，徐々に積み上げていくことが重要であると認識されている．

そのための手法として，節集合の3リテラル連言標準形（以下，3-CNF）への変換が考えられる．良く知られているように，任意の節集合は3-CNFへ変換可能である．本論文では文献 [ホップクロフト 80] の10.4節の変換法を使用する．長さnの節を変換すると$n-2$個の節が生成され，節の数は増えてしまうが，個々の節の長さは最大で3となり，短くなる．これにより個々のタブローを小さくできると考えられる．また変換に必要な計算の手間も線形オーダーで極めて小さい．**4・2**節で3-CNF変換手法の実験的評価を示す．

3・2 選択・交叉

本研究で考える選択・交叉の概略を述べる．

(1) 適応度の高い親タブローT_1をルーレット選択で選ぶ．次にその中で，置き換えるべき適応度が低い部分タブローST_1を確率的に選ぶ．

(2) 他のタブローの部分タブローを探し，ST_1の代わりに置き換えても全体がMEタブローの条件を満たすものを列挙し，その中から適応度が優れた部分タブローST_2を確率的に選ぶ．

(3) 実際にT_1からST_1を削除し，代わりにST_2を代入する．

通常GPでは交叉により二つの親遺伝子から二つの子を生成するが，本研究で1回の試行では一つの子を生成する．

3・3 適 応 度

前述の選択・交叉がうまく働くためには，適切な適応度を用いる必要がある．ここでは適応度を，親タブローT_1，その部分タブローST_1，ならびにST_1の代わりとなる部分タブローST_2の三つに分けて考える．

〔1〕 親タブローの適応度

親タブローT_1の適応度関数$\mathrm{F}_P(T_1)$は，以下では常に$\mathrm{F}_P(T_1) > 0$であり，0に近いほど良いと考える．

タブローの開道は長いほどリテラルを多く含むので，将来的には閉じる可能性が高いと考えられる．そこで開道の適応度として道の長さの逆数を取り，タブロー全体の適応度$\mathrm{F}'_P(T)$として，全ての開道の評価値の和を考える．即ち，Tの開道を$b_1 \cdots b_k$とするとき，

$$\mathrm{F}'_P(T) = \frac{1}{\mathrm{D}(b_1)} + \cdots + \frac{1}{\mathrm{D}(b_k)}$$

タブロー自体は小さいほど望ましい．しかし大きなタブローの方が内部の道が長いので，$\mathrm{F}'_P(T)$は良い値を取ることが多い．そこでタブローの大きさを制限し，制限値を超えた分はペナルティーとして評価値を減じることを考える．制限値は世代が進めば緩めていき，次第に大きなタブローの構成も許していく．遺伝子の世代gに無条件に許容できるタブローの大きさを$G(g)$とするとき，ペナルティー関数$\mathrm{P}(T)$を

$$\mathrm{P}(T) = \begin{cases} |T| - G(g) & \text{if } |T| > G(g) \\ 0 & \text{otherwise} \end{cases}$$

と定める．最終的に$\mathrm{F}_P(T)$を以下のように定める．

$$\mathrm{F}_P(T) = \mathrm{F}'_P(T) + \mathrm{P}(T)$$

〔2〕 親タブロー内の部分タブローの適応度

タブローT_1中の部分タブローST_1は，開ゴールを多く含むほど適応度が悪く，優先して置き換えるべきものと考えられる．そこでmをST_1に出現する親T_1の開ゴールの個数とするとき，T_1中のST_1の適応度$\mathrm{F}_C(ST_1)$を以下のように定める．

$$\mathrm{F}_C = \frac{m}{|ST_1|}$$

〔3〕 代入する部分タブローの適応度

ST_1の代わりとして別の優秀な部分タブローST_2を選択したい．ST_1をST_2に置き換えたときに閉タブローが生じる率は，ST_2自身の開ゴールのラベルの種類の数[*1]に反比例すると考えられる．但し前と同様に，開ゴールの数だけでST_2を選択すると，タブローが異常に大きくなる場合がある．タブローはで小さい方が望ましい．よって前述のペナルティー関数$\mathrm{P}(ST)$を併用する．ST_2選択のための適応度$\mathrm{F}_S(ST_2)$を次式で定義する．

$$\mathrm{F}_S(ST_2) = (ST_2\text{自身の開ゴールの種類数}) + \mathrm{P}(ST_2)$$

3・4 突 然 変 異

本研究では以下の2種類の突然変異を考える．

(1) 遺伝子の多様性の保持を目的とする通常の意味での突然変異．タブローの部分木をある確率でランダムに変更・生成する．

(2) タブロー遺伝子から冗長な部分を除去・修正するための突然変異．本研究では以下の冗長性を対象とする．

(a) 同一リテラルが道に二度以上出現する場合，その道は冗長である．葉に近い方の重複リテラルを頂上節の一部とする部分タブローを削除し，新しい部分木をランダムに生成する．

(b) 単位節公理（または後述の単位節補題）に包摂される節Cがタブローに出現するならば冗長である．Cを頂上節とする部分木を削除し，新しい部分木をランダム生成する．

3・5 部分構造の再利用：補題

冗長な探索を避け，より有益で大きな遺伝子を生成するために，本研究では補題機構を導入する．補題とは一度証明した部分論理式のことである．例えばMEタブロー型定理証明では，節集合S上のMEタブローの全ての開ゴールの論理和はある種の補題をなす（**3・1**節参照）．得られた補題節をデータベースに登録して，もとの公理節と同列に扱い，同一タブローの再計算の抑制を図ることができる．定理証明の分野では多くの研究がなされており，GPを用いた定理証明においてかなりの威力を発揮するものと考えられる．本研究では以下の補題機構を導入する．

(1) 大域的な補題（単位節補題，2リテラル節補題，補題テーブルを利用した新たな単位節補題の生成）

(2) 局所的な非単位節補題（C-reduction）

上の補題機構の詳細は紙面の都合上省略させて頂く．[Astrachan 94, Baumgartner 98, Iwanuma 97,

---

*1 タブロー中の開ゴール$N_1, \cdots, N_k$のラベル集合$\{\lambda(N_1), \cdots, \lambda(N_k)\}$の大きさのこと．

表1　実験環境と主なパラメータ

使用マシン	SGI O2 (R5000 64MByte)
使用メモリ	30MByte
CPU 時間制限	3 600 秒
個体数	800
突然変異率	0.01
初期タブローの深さ	5
世代 g のタブローの大きさの制限値 $G(g)$	$G(g) = 40 + (g \div 4)$

Iwanuma 98a, Letz 94, Moser 97] を参照されたい．

4．実　験　結　果

GP を用いた ME タブロー型定理証明プログラムを試作し，性能評価実験を行なった．ベンチマーク問題は The second DIMACS implementation challenge: 1992-1993 [Johnson 93] にある充足不可能問題を使用した．また代表的な命題論理証明プログラム SATO3, SATO1 および POSIT [Zhang 97] を使用して，比較実験を行なった*2．SATO3 は，現時点で世界最高速とされる命題論理証明プログラムの一つである．実験環境と代表的パラメータは表1のとおりである*3．

4・1　3-CNF 問題

3-CNF 問題である Aim シリーズに対する実験結果を表2に示す．表中の数字は証明に要した CPU 時間(秒)，“–” は制限時間オーバーを示し，“+m” はメモリ不足による証明失敗を示している．GM-pro は我々が試作した証明プログラムである．GM-pro の列の括弧の中の数字は，証明が求まった世代を示している．

Aim シリーズの各問題の命題変数の数は 50 から 200 個であり，節の数は変数の個数の 1.6 から 2 倍，即ち 80 から 400 である．けっして易しい問題ではない．特に命題変数が 200 個ある Aim-200 シリーズはかなり難しい [Zhang 97]．本研究での比較実験でも SATO1 や POSIT が証明に失敗している．GM-pro は実験的なシステムであるにもかかわらず，安定して良好な成績を残している．

表2　3-CNF 問題での実験結果および比較

Aim-problems	POSIT	SATO3	SATO1	GM-pro	
50-1_6-no-1	0.00	0.01	0.52	2.4	(2)
50-1_6-no-2	0.01	0.01	0.04	6.2	(5)
50-1_6-no-3	0.05	0.01	1.02	2.7	(3)
50-1_6-no-4	0.00	0.01	0.02	2.8	(3)
50-2_0-no-1	0.00	0.01	0.73	5.5	(6)
50-2_0-no-2	0.01	0.01	0.04	3.6	(4)
50-2_0-no-3	0.03	0.01	0.31	5.4	(6)
50-2_0-no-4	0.03	0.01	0.01	5.2	(5)
100-1_6-no-1	18.83	0.01	1717.9	21.9	(19)
100-1_6-no-2	39.89	0.01	–	18.4	(17)
100-1_6-no-3	67.41	0.02	188.0	17.4	(17)
100-1_6-no-4	4.87	0.02	–	16.1	(17)
100-2_0-no-1	123.58	0.01	2870.3	6.2	(6)
100-2_0-no-2	51.07	0.01	2831.6	20.1	(23)
100-2_0-no-3	22.84	0.01	1439.1	13.9	(12)
100-2_0-no-4	57.22	0.01	–	27.7	(26)
200-1_6-no-1	–	0.03	–	26.8	(27)
200-1_6-no-2	–	0.02	–	13.2	(12)
200-1_6-no-3	–	0.05	–	26.3	(29)
200-1_6-no-4	–	0.02	–	7.7	(8)
200-2_0-no-1	–	0.05	–	57.7	(61)
200-2_0-no-2	–	0.05	–	38.7	(40)
200-2_0-no-3	–	0.04	–	28.0	(27)
200-2_0-no-4	–	0.04	–	26.6	(27)

表3　各種の補題の比較実験

Aim-problems	No-lem	1-lem	2-lem	1&2	C-red
50-1_6-no-1	561.4	21.3	3.7	3.7	19.4
50-1_6-no-2	–	37.5	10.7	7.9	151.3
50-1_6-no-3	+m	15.6	5.3	2.7	–
50-1_6-no-4	420.1	20.7	3.9	2.5	34.6
50-2_0-no-1	929.7	30.7	18.6	15.1	37.0
50-2_0-no-2	658.2	16.6	11.8	5.7	–
50-2_0-no-3	–	31.0	10.1	8.5	85.8
50-2_0-no-4	–	32.4	11.6	12.6	54.0
100-1_6-no-1	+m	+m	23.3	23.3	+m
100-1_6-no-2	+m	+m	475.0	43.5	–
100-1_6-no-3	+m	+m	39.4	37.5	–
100-1_6-no-4	+m	551.4	126.0	27.3	44.5
100-2_0-no-1	201.5	25.7	8.9	7.7	+m
100-2_0-no-2	+m	225.1	44.6	30.8	363.9
100-2_0-no-3	–	30.5	27.0	26.5	+m
100-2_0-no-4	+m	–	+m	24.4	+m
200-1_6-no-1	+m	69.8	35.0	+m	+m
200-1_6-no-2	+m	+m	+m	18.0	+m
200-1_6-no-3	–	+m	+m	5274.4	+m
200-1_6-no-4	+m	24.3	17.0	7.3	+m
200-2_0-no-1	–	665.8	+m	+m	+m
200-2_0-no-2	+m	+m	120.0	112.9	+m
200-2_0-no-3	+m	+m	42.1	+m	298.3
200-2_0-no-4	+m	808.6	41.3	30.4	192.2

表3は個々の補題技術の実験結果である．最上段は

- No-lem：補題技法を全く用いない場合，
- 1-lem：単位節補題のみを用いる場合，
- 2-lem：2 リテラル補題のみを用いる場合，
- 1&2：単位節補題と 2 リテラル補題を併用する場合，
- C-red：C-reduction のみを用いる場合

を表している．表2の “GM-pro” の列は，上記の補題技術を全て用いた場合の結果である．各種の補題技術が相当の効果を挙げていることが分かる．

*2　確率的な探索手法を用いて充足可能性を判定するプログラムとして GSAT や WalkSAT [Hoos 98] が著名であるが，これらは不完全アルゴリズムであるために充足不可能性の判定には使用できない．そのため本研究の比較の対象から外してある．

*3　表1の値は Aim シリーズの問題を用いた予備実験を通して求めたものである．

表4　非3-CNF問題での実験結果および比較

Jnh-problems	POSIT	SATO3	SATO1	GM-pro	
				Normal	3-CNF
Solved	34	34	34	0	15 (29.6)
Time Over	0	0	0	0	6 (36.8)
Memory Over	0	0	0	34	13 (42.3)

4・2　非3-CNF問題

3-CNFではない問題としてDIMACSのJnhシリーズを取り上げ，評価を行なった．Jnhの各問題は，それぞれ100の命題変数を持ち，節の数は800，850，900のいずれかである．単位節は全く出現しない．総数34の充足不能問題に対して表4の実験結果を得た．紙面の都合上，個別の問題に対するデータは省略する．

従来型の証明プログラムPOSIT, SATO3, SATO1は，それぞれすべての問題を1CPU秒以内で解いた．残念ながらGM-proは34題すべてメモリオーバーを起こし，解は一つも得ることができなかった（表4のNormalの列）*4．幾つかの要因が考えられるが，最も大きな原因の一つとしてJnhの各節が長いことが考えられる．Jnhの節は平均約6個，最大で11個のリテラルを持つ．このためタブローが不必要に大きくなったと考えられる．

従来型のPOSIT, SATO3, SATO1は全てDavis-Putnamのアルゴリズムを基礎としている．単位リテラル規則を可能な限り活用し，節を順次簡約していく．この過程で付随的に冗長な節やリテラルが簡約されることも多い．GM-proは命題論理に特化させていないため，この種の機能が弱い．この差が出ているものと思われる．

この対策として，節集合の3-CNFへの変換（[ホップクロフト80]の10.4節参照）を試みた．結果は最右の列（3-CNF）に示したが，かなりの改善が見られる．注意して頂きたいのは，長さnの節を3-CNF形へ変換すると，$n-2$個の（長さ3の）節が生成される点である．節の個数は約3000強に増えたにもかかわらず，GM-proは半数近くの問題の証明に新たに成功している．3-CNF列の括弧内の数字は最終世代（タイムオーバー，メモリオーバーした場合は，その直前の世代）でのタブロー800個の大きさの平均である*5．

*4　個体数，突然変異率などのパラメータはAimシリーズの実験と全く同じである．メモリオーバーフロー回避のための個体数の制限などのパラメータ調整は，全く行なっていないことに注記させて頂く．

*5　Aim問題でのタブローの最終世代での大きさの平均は42.7である．なお証明に成功した閉タブローは，Jnhの方がAimよりも小さくなる傾向が観測されている．

5. まとめ

本論文ではデータを省略したが，平均適応度は世代進行に対して全般的に漸進的に改善されていくことが確かめられている．この意味からも定理の自動証明においてGPの技法がある程度有効であると考えられる．

一階論理への拡張は今後の最も大きな課題である．AimとJnhシリーズに対する実験結果から，(1) 命題変数の多さはさほど問題とはならない，(2) 長い節の取り扱いには困難が伴う，(3) 節の個数はさほど問題とはならない，以上のことがかなり明瞭となった．一般にエルブランの定理より，充足不可能な一階の節集合Sは，充足不可能でSの有限個の基礎例からなるある集合$Ground(S)$と同一視できる．$Ground(S)$の式は本質的に命題論理の記号式であり，式の長さはSのものと本質的に変わらないが，原子式（＝命題変数）と節の個数はかなり増える．よって本GP手法を一階論理へ拡張するにあたって，上の（1）と（3）の性質は有利に働くものと期待できる．(2)の問題は，継続述語（continuation）を導入すれば一階論理上でも3-CNFへの変換が行なえるので，ある程度解決できると予想している．残された最大の課題は，変数を含むタブローの適応度の合理的な定義と考えられる．

最後に，本論文の技法はあくまで一階論理への拡張を前提にしており，命題論理を目的としたものではない．試作プログラムにも命題論理固有の高速化手法は全く組込まれていないことを再度お断りさせて頂く．

謝　辞

本研究は一部，通信・放送機構（TAO）の援助による．

◇参　考　文　献◇

[Astrachan 94] Owen L. Astrachan: METEOR: Exploring Model Elimination Theorem Proving, J. Automated Reasoning, Vol.13, pp.283–296 (1994).

[Baumgartner 98] 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 (1998).

[Fuchs 97] M. Fuchs: Evolving Combinators, Proc. 14th Inter. Conf. on Automated Deduction, LNAI, Vol.1249, pp.416–430 (1997).

[Johnson 93] D.S.Johnson and M.A.Trick (eds.): The Second DIMACS Implementation Challenges, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (http://dimacs.rutgers.edu/challenges/) (1993).

[ホップクロフト 80] J.ホップクロフト，J.ウルマン（野崎，野下訳）：アルゴリズムの設計と解析II，サイエンス社 (1980).

[Hoos 98] H. H Hoos and T. Stützle: SATLIB - The Satisfiability Library, (http://aida.intellektik.informatik.th-darmstadt.de/ hoos/SATLIB/) (1998).

[伊庭 96] 伊庭 斉志: 遺伝的プログラミング, 東京電機大学出版局 (1996).

[Inoue 92] K. Inoue: Linear Resolution for Consequence Finding, Artif. Intell., Vol.56, pp.301-353 (1992).

[Iwanuma 97] K. Iwanuma: Lemma Matching for a PTTP-Based Top-Down Theorem Prover, Proc. 14th Inter. Conf. on Automated Deduction, LNAI, Vol.1249, pp.146-160 (1997).

[Iwanuma 98a] 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 98b] K. Iwanuma: I-THOP Ver 1.01, in: C.Suttner and G.Sutcliffe, The CADE-14 ATP System Competition, J. of Automated Reasoning (1998).

[北野 97] 北野 宏明 編: 遺伝的アルゴリズム 1,2,3 産業図書 (1993, 1995, 1997).

[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).

[Loveland 69] D.W. Loveland: A Simplified Format for the Model Elimination Theorem-Proving Procedure, J. ACM, Vol.16, No.3, pp.349-363 (1969).

[Moser 97] M. Moser, O. Ibens, R. Letz, J. Steinbach, C. Goller, J. Schumann and K. Mayr: SETHEO and E-SETHEO, J. Automated Reasoning, Vol.18, pp.237–246 (1997).

[McCune 92] W. McCune and L. Wos: Experiments in Automated Deduction with Condensed Detachment, Proc. 11th Inter. Conf. on Automated Deduction, LNAI, Vol. 607, pp.210–223 (1992).

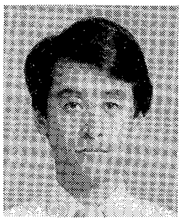
[中山 98] 中山, 藤田, 長谷川: GA-MGTP による Condensed Detachment 問題の解法, 信学技報, AI98-13 (1998).

[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).

[Zhang 97] H. Zhang Sato: An Efficient Propositional Prover, Proc. 14th Inter. Conf. on Automated Deduction, LNAI, Vol. 1249, pp.272-275 (1997).

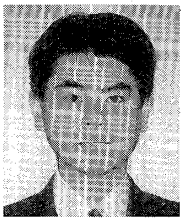
〔担当委員：阿久津達也〕

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. 現在山梨大学コンピュータメディア工学科助教授. 博士(工学). 人工知能の基礎論, 特に様相論理, 非単調論理, 定理自動証明プログラムの研究開発に従事. 1987,89,90,91 年人工知能学会全国大会優秀論文賞受賞.　〈iwanuma@esi.yamanashi.ac.jp〉



矢ヶ崎　剛史

1996 年 3 月山梨大学電子情報工学科卒業. 1998 年 3 月山梨大学大学院修士課程修了. 同年 4 月沖ソフトウエア（株）に入社. 主として定理自動証明と遺伝的アルゴリズムの研究に従事.