定理証明システム SENRI の構成†

山口高平†† 西岡弘明†††
打浪清一†† 手塚慶一††

従来の定理証明システムは，LISP で一つの定理証明法を構成することに重点が置かれていたが，使いやすさや効率面で問題があった．そこで本稿では，オプションの指定を変更するだけで，さまざまな定理証明法を構成でき，また反駁が得られない場合，支援システムが稼動し，適切な戦略を再設定できるように助言を与えることを新しい特徴とする定理証明システム SENRI (System to Evaluate Non-numeRical Informations) を構築した．次に LISP でシステムを構築すると，記憶領域の管理は廃品回収 (GC) を使用せざるをえないが，SENRI では，将来必要となる記憶領域は節集合領域だけであることに着目し，LAVS (List of AVailable Space) を節集合領域と一時作業領域に2分割し，両方向から使用してあふれが生じたとき，一時作業領域のポインタを再設定するだけで LAVS を再利用するという新しい記憶領域の管理法を提案し，GC と比較して LAVS の再構成時間が極度に短縮されることを確認した．また SENRI のリスト構造は，2進木リストに比べて，述語論理式をコンパクトに表現している．その他，SENRI の特徴としては，頻繁に使用されるルーチンのアセンブラ・バージョンを作成することにより1回当りの導出時間を短縮しており，また元バージョンはモジュール別に FORTRAN で作成したので，拡張性および移植性が高く，最後に表示は論理学で用いられる表記法に近いものを採用したので，入力が容易で出力が見やすいものになっていることが挙げられる．

1. まえがき

1965年に J. A. Robinson は導出原理を発表し，証明可能な定理は確実に導出法により証明できること（完全性）を示した．しかしながら，この方式は効率面で改善の余地があり，導出の適用範囲を制限し探索空間を狭める研究—「制限付導出法」および「戦略」の研究—が，数多くなされてきた．

一方，これらの成果を実働化する定理証明システムとしては，Chang の TPU[2] および山崎らのシステム[4]等があるが，これらは汎用記号処理言語 LISP で構成されており，一つの定理証明法（本稿では，制限付導出法は同じでも付加する戦略が異なれば，それらは異なった定理証明法として考える）を簡潔に記せることに重点が置かれ，異なった定理証明法を比較するのは困難であり，また反駁が得られない場合の支援等は考慮されていなかった．そこで本研究では，さまざまな定理証明法を容易に表現でき，また反駁が得られない場合，導出過程を分析して診断メッセージを出力することにより，適切な戦略を再設定できるように支援を行うことを新しい特徴とする定理証明システム SENRI を構築した．次に，SENRI のシステム内部では，データ構造としてリスト構造を節の表現に適した形式でデータ圧縮したものを採用しており，記憶領域の管理として GC を使用せずに，節集合領域と一時作業領域を LAVS において2分割し，両方向から使用してあふれが生じたとき，一時作業領域のポインタを再設定するだけで LAVS を再利用するという記憶領域の管理法を新しく採用し，LAVS の再構成時間を極度に短縮している．そのほか，SENRI では，使用頻度の高いルーチンは，アセンブラ・バージョンに置き換えることによって実行効率の改善を図り，また元バージョンは，モジュール別に FORTRAN で作成したので，拡張性が高いとともに移植性も高く，最後に，入出力は論理学で用いられる表記法に近い表示に合わせることによって使いやすくしている．

本稿の以下の部分では，まず SENRI のシステム構成について述べ，次に効率改善の経過を示し，最後に他のシステムと比較することにより，SENRI の有効性について検討する．

2. システム構成

2.1 システムの概観

SENRI の概観を**図1**に示す．SENRI では，まずユーザの選択した導出法と戦略および入力節集合を入力モジュール (RESOLUTION & STRATEGY SELECTOR, INPUTTER) によって読み込み，選択された制限付導出法実行モジュール(SNL EXECUTOR,

† The Organization of Theorem Proving System "SENRI" by TAKAHIRA YAMAGUCHI (Faculty of Engineering, Osaka University), HIROAKI NISHIOKA (Information Processing Center, Yamaguchi University), SEIICHI UCHINAMI and YOSHIKAZU TEZUKA (Faculty of Engineering, Osaka University).
†† 大阪大学工学部通信工学科
††† 山口大学情報処理センター

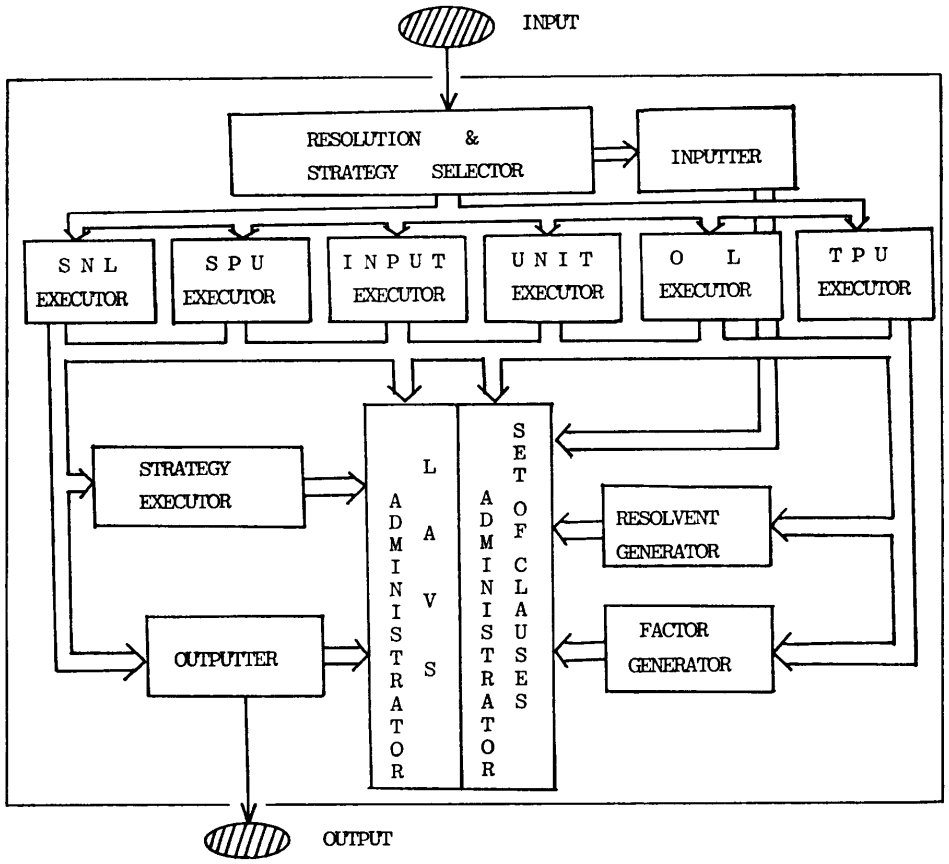


図 1 定理証明システム SENRI の構成
Fig. 1 The organization of theorem proving system "SENRI".

SPU EXECUTOR, INPUT EXECUTOR, UNIT EXECUTOR, OL EXECUTOR, TPU EXECUTOR) に制御が移る．以下，戦略モジュール (STRATEGY EXECUTOR) によって選択された戦略を適用しながら，推論モジュール (RESOLVENT GENERATOR, FACTOR GENERATOR) によって導出形 (resolvent) および簡約形 (factor) を生成し，空節 (empty clause) が導出されれば出力モジュール (OUTPUTTER) により反駁木 (refutation tree) を出力して停止する．

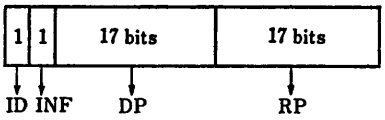
節リストは，生成順に連結され，節集合リストとして節集合管理モジュール (SET OF CLAUSES ADMINISTRATOR) によって管理されており，LAVS は LAVS 管理モジュール (LAVS ADMINISTRATOR) によって管理されている．また，セルの操作・文字データ領域の管理・リストコピー・代入操作とその応用操作・節の種類判定等のモジュールは図示されていないが，システムの随所で数多く使用されている．

2.2 データ構造

SENRI のデータ構造の基本単位であるセルは，実際には計算機の 1 ワード (36 bits) で構成されており，**図 2** に示す構造をもつ．入力された節は，**図 3** の具体例で示すとおり，セルを基本単位とするリスト構造によって表現される．

2.3 入力モジュール

SENRI における入力形式は，**図 4** の文法で示され，導出法・戦略・節集合の順に指定する．導出法と戦略は，ユーザのオプションとして**表 1** から選択できる．戦略には，支持集合の指定(1)，インタラクティブに最適値を決定する必要がないもの (2～4)，あるもの (5～7)，各導出法に専用のもの (8～12) があり，その他出力条件および停止条件に関するものとして 13～15 がある．戦略の各パラメータについては，おのおの暗黙値が存在するためユーザが指定しない場合には，この値が採られる．入力の具体例を**図 5**



ID部 → セルの識別情報をもつ.

セルの種類	値
リストセル	0
データセル	1

INF部 → データセルのとき，データの型情報をもつ.

データの型	値
定　数	0
変　数	1
関数記号	0
述語記号	0
補の述語記号	1

注) 同値のデータの型の区別は，セルの位置を情報としてソフトウェアで行う．また，枠付きリテラル(framed literal)を表現するときは，リストセルにおいて1をセットする.

DP部 → データセルのときは，文字データを指すためのポインタをもち，リストセルのときは，下連結のためのポインタをもつ.

RP部 → 右連結のためのポインタをもつ.

図 2 セルの構造
Fig. 2 Cell structure.

に示すとともに，**表 2** に本モジュールの概要を示す*.

2.4 制限付導出法実行モジュール

本モジュールは，表1に示す制限付導出法を実行するためにあり，表1の戦略において，1, 4, 6, 8, 9, 11, 13～15の処理は，このなかに組み込まれている．**表 3** に本モジュールの概要を示す.

* モジュールの概要は，モジュール構成のサブルーチンを用いて表し，以下の形式を用いる.

モジュール名	
サブルーチン名	サブルーチンの概要

2.5 推論モジュール

本モジュールは，導出形および簡約形を生成するためにあり，生成される導出形のリストと簡約形のリストは，節集合管理モジュールで統一的に処理される目的で，**図 6** と**図 7** に示すような同一構造をもつ．**表 4** に本モジュールの概要を示す.

2.6 戦略実行モジュール

本モジュールは，制限付導出法実行モジュールの中で取り扱われていない戦略を実行するためにあり，**表 5** にその概要を示す.

2.7 出力モジュール

本モジュールは，証明結果を出力するためにあり，**表 6** にその概要を示す.

2.8 節集合管理モジュール

本モジュールは，以下の四つの機能をもつ.

(1) 「初期設定」→入力節集合の最後の節に INPUT と LEND という二つの管理変数をセットし，INPUT は入力節の終りを示す変数として，LEND は節集合の終りを示す変数として，以後取り扱われる.

(2) 「導出形・簡約形の付加」→節集合リストの最後に，導出形リストおよび簡約形リストを連結する.

(3) 「節の探索」→節集合中の節の順位から，節のリストを取り出す.

(4) 「節の削除」→節集合中の最右の節を削除する.

本モジュールの概要を**表 7** に示す.

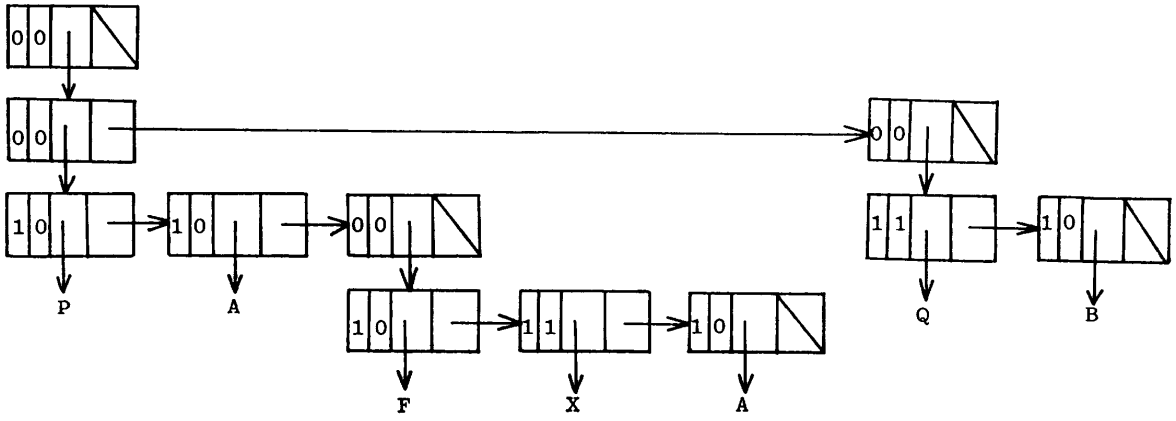


図 3 $P(a, f(x, a))V\sim Q(b)$ の内部構造
Fig. 3 Internal structure of "$P(a, f(x, a))V\sim Q(b)$".

```
<INPUT>::=<RESOLUTION>(<STRATEGY SEQUENCE>)<SET OF CLAUSES>.
<RESOLUTION>::= SNL|SPU|TPU|OL|INPUT|UNIT
<STRATEGY SEQUENCE>::=<STRATEGY>=<PARAMETER>|<STRATEGY>=<PARAMETER>,<STRATEGY SEQUENCE>
<STRATEGY>::= SUPPORT|TAUTO|EQRSLVNT|NOFACTOR|FDEPTH|RDEPTH|CLENGTH|TPUN2|TPUN3|TPUN4|
              TPUSUPPORT|OCLENGTH|ANSWER|NOTANS|PRINTRSLVNT
<PARAMETER>::= Y|N|<UNSIGNED INTEGER>|<SUPPORT SET>|<TPU SUPPORT SET>
<SUPPORT SET>::=(<ELEMENTS>)|()
<ELEMENTS>::=<UNSIGNED INTEGER>|<UNSIGNED INTEGER>,<ELEMENTS>
<TPU SUPPORT SET>::=(<TPU ELEMENTS>)
<TPU ELEMENTS>::=()|(),<TPU ELEMENTS>|(<ELEMENTS>)|(<ELEMENTS>),<TPU ELEMENTS>
<SET OF CLAUSES>::=<NON-EMPTY CLAUSE>|<NON-EMPTY CLAUSE>;<SET OF CLAUSE>
<NON-EMPTY CLAUSE>::=<LITERAL SEQUENCE>
<LITERAL SEQUENCE>::=<LITERAL1>|<LITERAL1><OR><LITERAL SEQUENCE>
<LITERAL1>::= #<LITERAL2>|<LITERAL2>
<LITERAL2>::=<ATOMIC FORMULA>|<NOT><ATOMIC FORMULA>
<ATOMIC FORMULA>::=<PREDICATE SYMBOL>(<TERM SEQUENCE>)|<PREDICATE SYMBOL>
<TERM SEQUENCE>::=<TERM>|<TERM>,<TERM SEQUENCE>
<TERM>::=<CONSTANT>|<VARIABLE>|<FUNCTION>
<FUNCTION>::=<FUNCTION SYMBOL>(<TERM SEQUENCE>)
<VARIABLE>::= $<IDENTIFIER>
<CONSTANT>::=<IDENTIFIER>
<PREDICATE SYMBOL>::=<IDENTIFIER>
<FUNCTION SYMBOL>::=<IDENTIFIER>
<OR>::= /
<NOT>::= -
```

[注意事項]
(1) 識別子の長さは任意であるが，9文字目以降は無視される．
(2) 入力は，自由欄形式である．
(3) 注釈は，/* 文字列 */の形式で，入力の任意の場所に挿入できる．ただし，文字列には */を含まない．

図 4 SENRI のシンタックス
Fig. 4 Syntax of SENRI.

表 1 導出法および戦略指定のためのユーザのオプション
Table 1 User's options to select resolutions and strategies.

	指定形式	指定内容
導出法	SNL	SNL 導出法の実行
	SPU	SPU 導出法の実行
	TPU	TPU システムのシミュレート
	OL	OL 導出法の実行
	INPUT	INPUT 導出法の実行
	UNIT	UNIT 導出法の実行
戦略	1 SUPPORT=(*m*,..)	支持集合の指定
	2 TAUTOLOGY=*Y*	恒真節の除去
	3 EQRSLVNT=*Y*	等位導出形の除去
	4 NOFACTOR=*Y*	簡約操作 (factoring) の除去
	5 FDEPTH=*m*	関数の深さを *m* に制限
	6 RDEPTH=*m*	導出の深さを *m* に制限
	7 CLENGTH=*m*	節の長さを *m* に制限
	8 TPUN 2=*m*	TPU システムにおいてパラメータ N2 を *m* に指定
	9 TPUN 3=*m*	TPU システムにおいてパラメータ N3 を *m* に指定
	10 TPUN 4=*m*	TPU システムにおいてパラメータ N4 を *m* に指定
	11 TPUSUPPORT=((*m*,..),..)	TPU システムにおいて支持集合の指定
	12 OCLENGTH=*m*	OL 導出法において非枠付きリテラル (non-framed literal) の数を *m* に制限
	13 ANSWER=*Y*	unit answering clause での停止要求
	14 NOTANS=*Y*	述語名が NOTANS の負単位節での停止要求
	15 PRINTRSLVNT=*Y*	生成された導出形の出力要求

```
SNL(NOFACTOR=Y,FDEPTH=3,RDEPTH=4,EQRSLVNT=Y,CLENGTH=3,PRINTRSL=Y)
/*     EXAMPLE OF GROUP THEORY (TPU 1)    */
P(G($X,$Y),$X,$Y) ; -P(K($X),$X,K($X)) ; P($X,H($X,$Y),$Y) ;
P($X,$V,$W)/-P($X,$Y,$U)/-P($Y,$Z,$V)/-P($U,$Z,$W) ;
P($U,$Z,$W)/-P($Y,$Z,$V)/-P($X,$Y,$U)/-P($X,$V,$W) .
```

図 5 SENRI の入力例
Fig. 5 Input example of SENRI.

表 2 入力モジュールの概要
Table 2 Input module.

RESOLUTION & STRATEGY SELECTOR	
TPINPT(IR, IST, IS, ISUP, IPNT)	ユーザの指定した導出法と戦略を読み込む．IR と IST にそれらの値をセットし，IS には以下に示す SCLIN を呼び出し，その引数値がセットされる．支持集合の指定がある場合は，ISUP(TPU を実行するときは IPNT に支持集合の管理情報)にその値をセットする．
INPUTTER	
CLIN(N)	入力節を節リストに変換して，その番地を N にセットする．
SCLIN(N)	CLIN によって節を読み込み，節集合リストを生成し，その番地を N にセットする．

表 3 制限付導出法実行モジュールの概要
Table 3 The module to execute restricted resolutions.

SNL EXECUTOR	
SNL (IST, IS, ISP)	戦略 IST (支持集合の指定は ISP) の下で，入力節集合 IS から SNL 導出法を実行する．
SPU EXECUTOR	
SPU (IST, IS, ISP)	戦略 IST (支持集合の指定は ISP) の下で，入力節集合 IS から SPU 導出法を実行する．
TPU EXECUTOR	
TPU (IST, IS, ISP, IPNT)	戦略 IST (支持集合の指定は ISP，支持集合の管理情報は IPNT) の下で，入力節集合 IS から TPU システムを実行する．
OL EXECUTOR	
OL (IST, IS, ISP)	戦略 IST (支持集合の指定は ISP) の下で，入力節集合 IS から OL 導出法を実行する．
INPUT EXECUTOR	
INPUT (IST, IS, ISP)	戦略 IST(支持集合の指定は ISP)の下で，入力節集合 IS から INPUT 導出法を実行する．
UNIT EXECUTOR	
UNIT (IST, IS, ISP)	戦略 IST(支持集合の指定は ISP)の下で，入力節集合 IS から UNIT 導出法を実行する．

2.9 LAVS 管理モジュール

SENRI ではデータ構造としてリスト構造を採用しているため，記憶領域を多く必要とする問題では，LAVS の再構成が頻繁に発生してしまい，全体の実行時間に大きく影響してしまう．そこで，本モジュールは，図 8 に示すように LAVS を将来必要となる記憶領域（節集合領域）とそうでない記憶領域（一時作業領域）に 2 分割を行って両方向から使用し，あふれが生じたとき，一時作業領域のポインタを初期設定することにより，一時作業領域を再利用するという LAVS の管理を行っている．また，バックトラッキングが発生した場合は，それに応じて節集合領域のポインタを再設定している．本モジュールの概要を表 8 に示す．

3. SENRI の効率改善

抽象代数等の数学理論の多くは，Horn 節集合のクラスの問題として表現できる．したがって，本章で

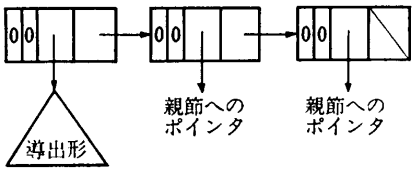


図 6 導出形のリスト
Fig. 6 List of resolvent.

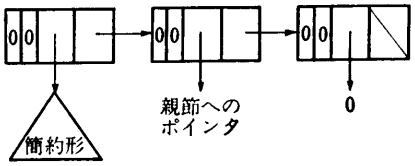


図 7 簡約形のリスト
Fig. 7 List of factor.

表 4 推論モジュールの概要
Table 4 Inference module.

RESOLVENT GENERATOR	
RESOLV (LRESLV, IFLAG, ICC 1, N 1, ICC 2, N 2)	節 ICC 1 の N 1 番目のリテラルと節 ICC 2 の N 2 番目のリテラルとから，一般の導出による導出形リストを生成する．非空節を生成した場合は，IFLAG に 1 をセットし，LRESLV にその番地をセットする．空節を生成した場合は，IFLAG に 0 をセットし，LRESLV にその番地をセットする．導出不可能の場合は，IFLAG に −1 をセットし，対応するリテラルが存在しない場合は，IFLAG に −2 をセットする．
SNLRSV (LRESLV, IFLAG, ICC 1, ICC 2, N 2)	節 ICC 1 の最右リテラルと節 ICC 2 の N 2 番目のリテラルとから，SNL 導出による導出形リストを生成する．LRESLV と IFLAG には，RESOLV と同様の値がセットされる．
OLRSLV (LRESLV, IFLAG, ICC 1, ICC 2, N 2, NLC)	節 ICC 1 の最右リテラルと節 ICC 2 の N 2 番目のリテラルとから，OL 導出による導出形リストを生成する．LRESLV と IFLAG には，RESOLV と同様の値がセットされ，NLC には簡約形を生成するときの制御情報がセットされる．
FACTOR GENERATOR	
INFACT (IC)	入力節 IC の簡約形を生成し，それを入力節集合に加える．
OINFCT (IC)	INFACT (IC) と同様であるが，OL 導出を実行するときに使用される．
RSFACT (IFACT, IC, NONNEW, NO)	導出形 IC から生成される簡約形の集合から，NO 番目の簡約形を取り出し，IFACT にその番地をセットする．簡約形が生成されない場合は，IFACT に 0 がセットされる．NONNEW は導出形の nonnew literal[9] の数である．
ORFACT (IFACT, IC, NLC, NO)	NLC が簡約形を生成するときの制御情報である以外は，RSFACT (IC) と同様であり，OL 導出を実行するときに使用される．

表 5 戦略実行モジュールの概要
Table 5 The module to execute strategies.

STRATEGY EXECUTOR	
ITAUTO (IC)	節 IC が恒真節ならば関数値を 1 とし，そうでなければ 0 とする．ただし，OL 導出法を実行する場合は，枠付きリテラル (framed literal) の取扱いが必要であるため，IOTAUT (IC) を用いる．
IEQ (IC1, IC2)	節 IC 1 と節 IC 2 が等しければ関数値を 1 とし，そうでなければ 0 とする．ただし，OL 導出法を実行する場合は，IOEQ (IC 1, IC 2) を用いる．
LENGTH (IC)	節 IC の長さを関数値とする．OL 導出法を実行する場合は，IOLENG (IC) を用いる．
IDEPTH (IC)	節 IC の関数の深さを関数値とする．

は，ほとんどの問題が Horn 節集合となる TPU EXAMPLE 1〜9[1] を証明問題として選び*，SNL および SPU 導出法をそのまま実行する SENRI FORTRAN バージョンから，徐々に効率改善を図った経過を示す．

最初のバージョンで実験した効率の結果は，表 10 (A)に示されるが，このバージョンでは，無駄な導出が頻繁に発生し，多くの問題が証明未完了である．そこで，効率改善に影響を与える要因を分析し，次の 3 項目に着眼し改善を図った（**図 9** 参照）．

（1） 戦略と制限付導出法 (SNL・SPU) を組み合わせることにより，探索空間を狭める．

（2） 頻繁に使用するルーチンのアセンブラ・バージョンを作成することにより，1 回当りの導出時

* TPU EXAMPLE 9 だけは renaming[9] の操作を加えても Horn 節集合に変換されない．以後 TPU EX 1〜9 と略記する．

表 6 出力モジュールの概要
Table 6 Output module.

OUTPUTTER	
CLOUT (IC)	節 IC を出力する.
DTREE (IC)	節 IC の演繹木を出力する. IC が空節ならば, 証明結果を出力することになる.

表 7 節集合管理モジュールの概要
Table 7 The module for administration of set of clauses.

SET OF CLAUSES ADMINISTRATOR	
LCSET (C, L)	節集合の管理を行い, 引数Cによってその機能と関数値は以下のようになる.

C	機能	関数値	C	機能	関数値
'S'	初期設定	LEND	'G'	節の探索	L 番目の節の番地
'P'	導出形の付加	LEND	'D'	節の削除	0

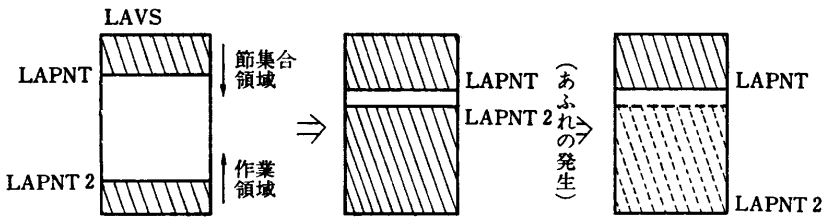


図 8 SENRI における LAVS の管理法
Fig. 8 The administration of LAVS in SENRI.

表 8 LAVS 管理モジュールの概要
Table 8 LAVS administration module.

LAVS ADMINISTRATOR	
GETCEL (L)	LAVS の上位からセルを取り出す.
GETCL 2 (L)	LAVS の下位からセルを取り出す.
BACKT	バックトラッキングが発生したとき, ポインタを再設定する.

間を短縮する.

(3) 記憶領域の管理に, GC を使用せずに 2.9 節に示した手法を用いることにより, LAVS の再構成時間を短縮する.

表 9 に各効率実験モードを示すとともに, **表 10** (A)～(D)にその実験結果を示す. モード 2 では, 制限付導出法に戦略を付加した結果, 探索空間が狭められ, より多くの問題が証明されうることを示している. モード 3 では, さらにアセンブラ・バージョンにより導出を実行しているため, 1 回当りの導出時間が短縮され, モード 2 と比較して, 2.6～4.5 倍程度効率が改善されている. さらにモード 4 では, LAVS の再構成に GC を使用せずに, 2.9 節で示した手法を用いることにより, GC が起動される問題 (TPU EX-1, 3, 4, 9) では, モード 3 と比較してほぼ 2 倍程度効率が改善されている. また, GC の起動のタイミングの影響により, SPU 導出法で証明未完了であった TPU EX-2, 3 も証明を完了している. モード 4 で効率改善されたのは, GC が起動した場合, 自由領域を回復するのに, 将来必要となるセルのマーク付けおよび将来必要とならないセルの再構成という大きな実行コストを要する作業を行わなければならないのに対し, 2.9 節で示した手法では, たんに一時作業領域のポインタを再設定するだけで済むためである. ただし, バックトラックが生じたときは, 節集合領域のポインタも後戻りさせている.

表 9 各効率実験モードの効率改善項目設定
Table 9 The mode of experiment for efficiency.

効率改善項目 / モード	探索空間の狭化法	システムモジュール	記憶領域の管理法
Mode 1	SNL, SPU	FORTRAN version	Garbage collector
Mode 2	SNL, SPU+strategies	FORTRAN version	Garbage collector
Mode 3	SNL, SPU+strategies	Assembler version	Garbage collector
Mode 4	SNL, SPU+strategies	Assembler version	LAVS administrator

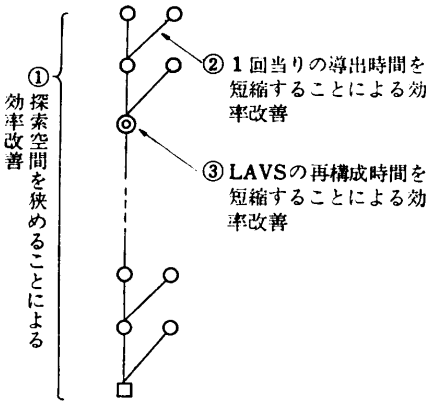


図 9 効率改善のための三つの要因
Fig. 9 Three factors for improving efficiency.

最終モードでは，本実験で用いたほとんどの問題が数秒以内で証明を完了しており，実用的なシステムになったと考えられる．

4. システム評価

4.1 SENRI と LISP で構成したシステムとの比較評価

多くの定理証明システムは，汎用記号処理言語 LISP で構成されており，SENRI を評価する場合 LISP で構成された他のシステムとの比較は欠かせない．そこで，LISP で構成された TPU システムと同じ導出法と戦略を SENRI で構成し，比較検討を行った．

4.1.1 データ構造の比較

SENRI の基本データ構造は，LISP のものと比較して，INF 部により述語論理式をコンパクトに表現している．具体的には，変数および補の述語記号の取扱いに関して，LISP で構成された多くのシステムでは特別な部分リストを設けるのに対して，SENRI では INF 部に 1 をセットし，ソフトウェアで管理できる．たとえば**図 10** では，LISP における X と Y の変数部分リストが，SENRI においては，X と Y のデータセルの INF 部に 1 をセットすることで表現されている．また，NOT のデータセルは，P のデータセルの INF 部に 1 をセットすることで表現されている．

本データ構造は，変数および補の述語記号を INF 部で表現することによって，変数と定数および正の述語記号と負の述語記号を含むリテラルリスト構造が対称的になり，リスト構造と比べて記憶効率が改善された新しいデータ構造[12]を提案する基礎にもなっている．

表 10 (A) モード 1 の実行効率
Table 10 (A) The execution efficiency of mode 1.

Problem \ Resolution	SNL	SPU
TPU EX-1	*	2924 (m sec)
TPU EX-2	*	Retired
TPU EX-3	*	Retired
TPU EX-4	*	Retired
TPU EX-5	183 (m sec)	7230
TPU EX-6	229	Retired
TPU EX-7	*	1577
TPU EX-8	*	Retired
TPU EX-9	15431	Rejection

*: LAVS empty or Stack overflow
Retired: More than 30 sec
LAVS: 10,000 cells (36 bits/cell)
Machine: ACOS system 1000 model 40

表 10 (B) モード 2 の実行効率
Table 10 (B) The execution efficiency of mode 2.

Problem \ Resolution	SNL	SPU
TPU EX-1	1286 (m sec)	785 (m sec)
TPU EX-2	Retired	*
TPU EX-3	1117	*
TPU EX-4	1214	7732
TPU EX-5	31	1286
TPU EX-6	65	5572
TPU EX-7	126	287
TPU EX-8	330	1445
TPU EX-9	937	Rejection

表 10 (C) モード3の実行効率
Table 10 (C) The execution efficiency of mode 3.

Problem \ Resolution	SNL	SPU
TPU EX-1	400 (m sec)	296 (m sec)
TPU EX-2	Retired	*
TPU EX-3	369	*
TPU EX-4	387	2912
TPU EX-5	8	399
TPU EX-6	15	2082
TPU EX-7	28	67
TPU EX-8	76	597
TPU EX-9	356	Rejection

表 10 (D) モード4の実行効率
Table 10 (D) The execution efficiency of mode 4.

Problem \ Resolution	SNL	SPU
TPU EX-1	212 (m sec)	119 (m sec)
TPU EX-2	Retired*	5661
TPU EX-3	179	4005
TPU EX-4	196	1162
TPU EX-5	7	216
TPU EX-6	14	853
TPU EX-7	25	59
TPU EX-8	69	223
TPU EX-9	157	Rejection*

* SNL 導出法で TPU EX-2 の証明を終えていないのは，この問題が縦型探索には，まったく適していないためだと考えられる．また，SPU 導出法で TPU EX-9 の証明が拒絶されているのは，この問題が Horn Set でないことに起因しており，SNL 導出法では偶然証明されたにすぎない．

4.1.2 入出力の比較

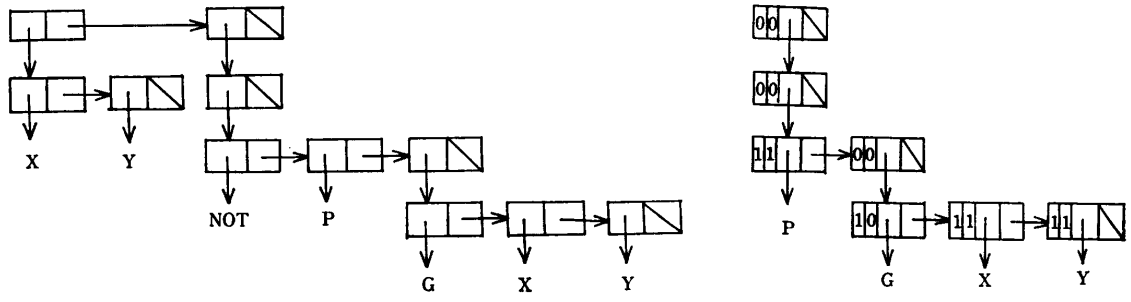
TPU EX-1 を例にとり，元データおよび LISP と SENRI における入出力形式を**図 11～15** に示す．図 12 と図 13 を比較することにより，SENRI* のほうが元データにより近い形式で入力できることがわかる．また，図 14 と図 15 を比較することにより，LISP ではたんに節番号を出力する形式であるのに対して，SENRI では反駁に用いられた節をすべて出力しており，導出の経過が容易に理解できることがわかる．

4.1.3 効率の比較

表 11 に TPU EX-1～9 を SENRI アセンブラ・バージョンで構成した TPU システムで証明したときの実行効率の結果を示す．LISP コンテスト[7]において，同問題を各種 LISP で実行した効率結果が報告されている．表 11 とこれらの結果は，ハードウェアの差異等で直接比較はできないが，本システムは現在最良と考えられる LISP コンパイラと同程度であり，サブルーチンのアセンブラ・バージョンをさらに増加したり，開いたサブルーチンを導入すれば，SENRI の効率改善は，よりいっそう進むと考えられる．

次に，SENRI FORTRAN バージョンに一括型の Garbage collector と LAVS administrator を導入したときの効率比較実験を行うことによって，LAVS の再構成時間について検討する．**表 12** と**図 16** より，LAVS administrator を導入したときのほうが，Garbage collector を導入したときより常に効率がよく，LAVS のサイズが小さいほどそのことが顕著に現れている．すなわち，Garbage collector を用いた場合，将来必要となるセルをマークし，将来必要とならないセルを再構成することにより，自由領域を回復するため，1回当りの LAVS の再構成時間が大きくなる．したがって，LAVS のサイズが小さいとき，

* LISP と SENRI では処理方式が異なるため，TPU システムの実行をシミュレーションするためには，入力節集合の順序を変更しなければならないことがある．



LISP による論理式の内部表現　　SENRI による論理式の内部表現

図 10 LISP と SENRI による $\sim P(g(x, y))$ の内部構造
Fig. 10 Internal structures of $\sim P(g(x, y))$ by LISP and SENRI.

(1) $P(g(x,y),x,y)$

(2) $P(x,h(x,y),y)$

(3) $\sim P(k(x),x,k(x))$

(4) $\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(x,v,w) \vee P(u,z,w)$

(5) $\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(u,z,w) \vee P(x,v,w)$.

図 11 TPU EXAMPLE 1 の元データ
Fig. 11 The original data of TPU EXAMPLE 1.

```
(TPU
  @((1(X Y)((P(G X Y)X Y)))
    (2(X Y)((P X(H X Y)Y))))
  @((3(X)((NOT P(K X)X(K X)))))
  @((4(X Y Z U V W)((NOT P X Y U)(NOT P Y Z V)(NOT P X V W)(P U Z W)))
    (5(X Y Z U V W)((NOT P X Y U)(NOT P Y Z V)(NOT P U Z W)(P X V W))))
  @((3)NIL)
  @5
  @2
  @3
  @0)
```

図 12 LISP の入力形式
Fig. 12 Input format of LISP.

起動回数が多くなり，ほとんどの時間が GC に費やされ，全体の効率が極端に悪くなる．一方，LAVS administrator を用いた場合，一時作業領域のポインタをたんに再設定するだけで，自由領域を回復できるため，たとえ LAVS のサイズが小さく，起動回数が多くなっても，LAVS の再構成時間は，ほとんど変化しない．したがって，LAVS のサイズが大きくとれないとき，とくに有効性を発揮すると考えられる．

4.2 他の定理証明システムとの比較評価

従来発表されてきたシステム[3),4)]は，一つの定理証明法を簡潔に記せることが主目標であり，異なった定理証明法を比較するには，その数だけプログラムを組まなければならずユーザへの負担が多かった．一方，SENRI では，異なった定理証明法を容易に比較できることが特色であり，表1に示したオプションの指定をたんに変更するだけで，定理証明法を変更できる．また，戦略は左から順に適用されるため，戦略のオプションは同じでも指定順序が違えば異なった定理証明法を表現することになり，定理証明法の詳細な変更が可能である．次に，SENRI は，モジュール別に構成されているので，拡張性が高いとともに，元バージョンは，FORTRAN で作成されているので，移植性も高い．最後に，以前のシステムでは，戦略の不適切に

```
((6 3 4 4)(11 2 6 2)(15 1 11 1)(CONTRADICTION 1 15)).
```

図 14 LISP の出力形式
Fig. 14 Output format of LISP.

表 11 TPU システムを実行する SENRI アセンブラバージョンの効率
Table 11 The efficiency of SENRI assembler version to execute TPU system.

Problem \ Resolution	TPU (m sec)
TPU EX-1	131
TPU EX-2	645
TPU EX-3	281
TPU EX-4	279
TPU EX-5	49
TPU EX-6	1,715
TPU EX-7	436
TPU EX-8	318
TPU EX-9	206

表 12 Garbage collector および LAVS administrator を使用したときの TPU システムを実行する SENRI FORTRAN バージョンの効率
Table 12 The efficiency of SENRI FORTRAN version to execute TPU system using Garbage collector and LAVS administrator.

cells \ LAVS reconstructor	Garbage collector (m sec)	LAVS administrator (m sec)
2,000	3,400	909
4,000	1,682	911
6,000	1,588	908
8,000	1,522	906
10,000	1,561	920

注) 実行時間は，TPU EX-1～9 の平均実行時間を示す．ただし，TPU EX-6 は，2,000 セルでは証明未完了であるため，除いた．

```
TPU(NSUPPORT=((3),()),TPUN2=2,TPUN3=3,TPUN4=0)
/*     EXAMPLE OF GROUP THEORY (TPU 1)     */
P(G(¥X,¥Y),¥X,¥Y) ; -P(K(¥X),¥X,K(¥X)) ; P(¥X,H(¥X,¥Y),¥Y) ;
P(¥X,¥V,¥W)/-P(¥X,¥Y,¥U)/-P(¥Y,¥Z,¥V)/-P(¥U,¥Z,¥W) ;
P(¥U,¥Z,¥W)/-P(¥Y,¥Z,¥V)/-P(¥X,¥Y,¥U)/-P(¥X,¥V,¥W) .
```

図 13 SENRI の入力形式
Fig. 13 Input format of SENRI.

```
+++ DEDUCTION +++

        *** INPUT CLAUSE(    1) ***
P(G(¥X,¥Y),¥X,¥Y)

        *** INPUT CLAUSE(    4) ***
-P(¥X,¥Y,¥U) / -P(¥Y,¥Z,¥V) / -P(¥X,¥V,¥W) / P(¥U,¥Z,¥W)

        *** INPUT CLAUSE(    3) ***
-P(K(¥X),¥X,K(¥X))

        *** RESOLVENT(    6) OF (    4) AND (    3) ***
-P(¥X1,¥X2,K(¥X3)) / -P(¥X2,¥X3,¥X4) / -P(¥X1,¥X4,K(¥X3))

        *** INPUT CLAUSE(    2) ***
P(¥X,H(¥X,¥Y),¥Y)

        *** RESOLVENT(   11) OF (    6) AND (    2) ***
-P(¥X1,¥X2,K(H(¥X2,¥X3))) / -P(¥X1,¥X3,K(H(¥X2,¥X3)))

        *** RESOLVENT(   15) OF (   11) AND (    1) ***
-P(G(¥X1,K(H(¥X1,¥X2))),¥X2,K(H(¥X1,¥X2)))

        *** RESOLVENT(   26) OF (    1) AND (   15) ***
EMPTY


+++ TPU SYSTEM END +++
```

図 15 SENRI の出力形式
Fig. 15 Output format of SENRI.

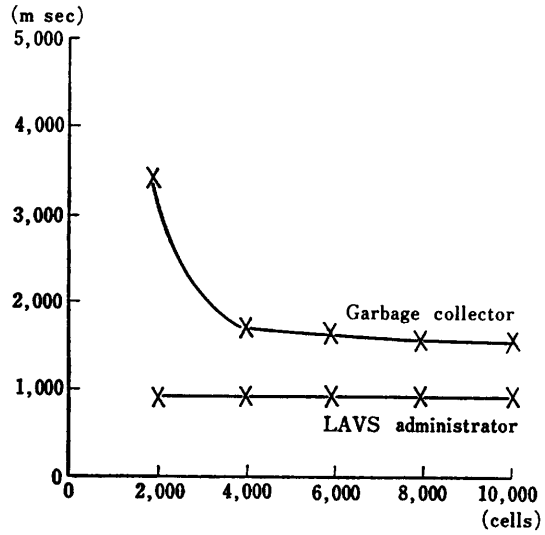


図 16 Garbage collector および LAVS administrator を使用したときの TPU システムを実行する SENRI FORTRAN バージョンの効率比較
Fig. 16 The efficiency comparison of SENRI FORTRAN version to execute TPU system using Garbage collector and LAVS administrator.

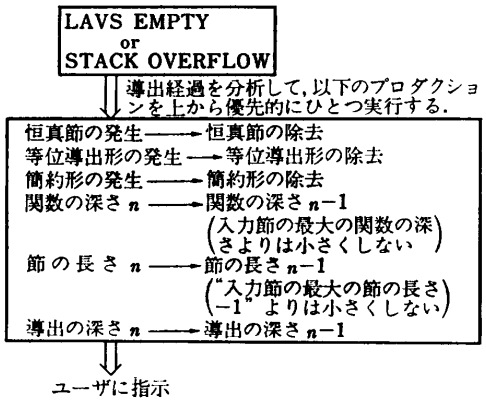


図 17 適切な戦略設定を行うための SENRI 支援システム
Fig. 17 Aid system of SENRI to set more adequate strategies.

より反駁が得られない場合，戦略の変更をユーザに一任していたが，SENRI では，**図 17** に示す支援システムによって導出過程を分析することにより，適切な戦略再設定の支援を実現している（**図 18** 参照）．

5. むすび

本研究では，定理証明システム SENRI を開発し

```
                *** THEOREM PROVER SENRI SYSTEM START ***


          +++ SNL RESOLUTION START +++


          +++INPUT CLAUSES+++

                    *** INPUT CLAUSE(    1) ***
            P(G(¥X,¥Y),¥X,¥Y)

                    *** INPUT CLAUSE(    2) ***
            -P(K(¥X),¥X,K(¥X))

                    *** INPUT CLAUSE(    3) ***
            P(¥X,H(¥X,¥Y),¥Y)

                    *** INPUT CLAUSE(    4) ***
            P(¥X,¥V,¥W) / -P(¥X,¥Y,¥U) / -P(¥Y,¥Z,¥V) / -P(¥U,¥Z,¥W)

                    *** INPUT CLAUSE(    5) ***
            P(¥U,¥Z,¥W) / -P(¥Y,¥Z,¥V) / -P(¥X,¥Y,¥U) / -P(¥X,¥V,¥W)

                                        |
                                        |
                                        |

                          *** FACTOR(   11) OF (    5) ***
                  P(¥W,¥Z,¥W) / -P(¥V,¥Z,¥V) / -P(¥X,¥V,¥W)

                          *** FACTOR(   12) OF (    6) ***
                  P(¥W,¥W,¥W) / -P(¥W,¥W,¥W)

                                        |
                                        |
                                        |

            +++RESOLVENT+++

                     *** RESOLVENT(   18) OF (    2) AND (    4) ***
              -P(K(¥X1),¥X2,¥X3) / -P(¥X2,¥X4,¥X1) / -P(¥X3,¥X4,K(¥X1))

                                        |
                                        |
                                        |

        *** RESOLVENT(   32) OF (   31) AND (    1) ***
-P(K(¥X1),G(¥X2,G(¥X3,G(¥X4,G(¥X5,G(¥X6,¥X1))))),G(¥X2,G(¥X3,G(¥X4,G(¥X5,G(¥X6,K
(¥X1)))))))
         * LAVS EMPTY AT GETCEL SUBROUTINE *

                +++ AID SYSTEM START +++


                * OCCURENCE OF TAUTOLOGY *
                * ADD STRATEGY TO DELETE TAUTOLOGY *


                +++ AID SYSTEM END +++
```

図 18 SENRI 支援システムの実行例
Fig. 18 An execution example of aid system in SENRI.

ていくうえで，まず内部的には，データ構造およびLAVSの再構成法に特色をもたせた．SENRIのデータ構造は，リスト構造を基礎にして，述語論理式(節)の表現に適するようにデータ圧縮したものを採用したが，リスト構造を基礎にしたのは，他のデータ構造と比べて，変更・修正等の操作が柔軟にでき，支援システム等の作成が容易になるからである．また，LAVSの再構成法は，最初GCを使用していたが，定理証明においては，将来必要となるセルは節集合の表現に関連したものだけであることに着目し，LAVSを双方向から利用して，将来必要とならない記憶領域をポインタの再設定だけで再利用するというLAVSの管理法を新しく提案し，GCと比較してその有効性を確認した．

次に外部的には，従来のシステムが，一つの定理証明法を簡潔に構成することに重点が置かれていたのに対して，SENRIでは，オプションの指定をたんに変更するだけで，さまざまな定理証明法を構成できるとともに，戦略の不適切により反駁が得られない場合，ユーザに助言を与えることを新しい特色としている．前者の特色からは，定理証明法の変更が容易となり，証明問題に対して実行効率のよい定理証明法を短時間で設定できるとともに，異なった定理証明法の比較が容易になる．また，後者の特色からは，適切な戦略設定が容易になる．

SENRIは，その他の特色として，頻繁に使用されるルーチンのアセンブラ・バージョンを作成して実行効率の改善を図り，また元バージョンは，FORTRANでモジュール別に作成したので，移植性および拡張性が高く，入出力は論理学で用いられる表記法に近いため，使いやすくなっていることが挙げられる．

以上のことから，定理証明システムSENRIは，従来のシステムと比べて，さまざまな定理証明法を短時間で構成できるとともに高速で使いやすいシステムになったと考えられる．また，近年研究が盛んなPROLOG[13]システムは，SNL導出法[5]の動作をプログラムの実行過程（計算法）と見立てるものと考えられるが，SENRIでは，それ以外の計算法も容易に構成できるため，効率のよい計算法を開発していく上で役立つと考えられる．

謝辞 ご討論いただいた手塚研究室の諸氏に厚く感謝します．

参考文献

1) Chang, C. L. and Lee, R. C.: *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York (1973).
2) Chang, C. L.: The Unit Proof and Input Proof in Theorem Proving, *J. Assoc. Comput. Mach.*, Vol. 17, No. 4, pp. 698-707 (1970).
3) Henschen, L., Overbeek, R. and Wos, L.: A Theorem-Proving Language for Experimentation, *Comm. ACM*, Vol. 17, No. 6, pp. 308-314 (1974).
4) 山崎，山本：定理の自動証明のためのプログラミングシステムの構成，情報処理，Vol. 7, No. 5, pp. 410-416 (1976).
5) Kuehner, D.: Some Special Purpose Resolution Systems, *Mach. Intell.*, Vol. 7, pp. 117-128 (1972).
6) Henschen, L. and Wos, L.: Unit Refutations and Horn Sets, *J. Assoc. Comput. Mach.*, Vol. 21, No. 4, pp. 590-605 (1974).
7) 竹内郁雄：第2回LISPコンテスト，情報処理，Vol. 20, No. 3, pp. 192-199 (1979).
8) 山口，西岡，打浪，手塚：拡張された定理証明システムSENRIについて，昭54信学部全大，p. 41 (1979).
9) 山口，西岡，打浪，手塚：定理証明システムSENRIのSNL導出法への拡張について，昭55信学総全大，p. 1200 (1980).
10) 山口，西岡，打浪，手塚：定理証明システムSENRIとその応用，信学技報，EC 80-26 (1980).
11) 山口，西岡，打浪，手塚：定理証明システムSENRIの構成，人工知能と対話技法，22-1 (1981).
12) 山口，阿部，打浪，手塚：論理プログラミングシステムSENRI/LPの内部構造について，昭57信学総全大，p. 1485 (1982).
13) Kowalski, R: *Logic for Problem Solving*, North-Holland, Amsterdam (1979).

(昭和57年10月18日受付)
(昭和58年7月19日採録)