北海道大学工学部研究報告
第83号（昭和52年）

Bulletin of the Faculty of Engineering,
Hokkaido University, No. 83 (1977)

検証条件のある証明システム“PROVER”について

柳　繁*　浦崎道教**　佐藤義治***　河口至商***

（昭和51年9月30日受理）

On a Proving System, PROVER, for the Verifivation Conditions

Shigeru YANAGI, Michinori URASAKI, Yoshiharu SATO and Michiaki KAWAGUCHI

(Reccived September 30, 1976)

Abstract

The problem of proving a program correctness was reduced to prove a set of formulae which are called verification conditions.

The purpose of this paper is to construct a system named PROVER which is designed for proving the verification conditions of program correctness, and is intended to produce a program verifying system.

In order to represent the verification conditions, we must introduce new functions. But there is no formal way to prove the formulae with the functions. And to prove these formulae, we have set forth the properties of new functions by certain adequate formulae which are interpreted as tautologies. Then these formulae are added to the deductive system of the PROVER as axioms.

By adopting this method, the PROVER increases its capacity for proviog formulae. And thus it is possible to construct a system which verifies the program correctness.

1. まえがき

プログラムの作成の際，“プログラムが正しく計算するか”という問題は重要である。この問題は，プログラムの正当性の問題として定式化される。従来，プログラムの正当性の検証は，テストデータを入力してデバックする事に留まった。その為，違いを発見するには有効であるが，プログラムに対する積極的な肯定とはなり得なかった。この正当性の検証は，近年のプログラム理論の発展によって，形式的な証明が可能となった。更に，プログラムの正当性の検証を電子計算機で自動的に行う事も試みられている。本研究の目的は，電子計算機によってプログラムの正当性を自動的に検証するシステムを試作する事である。

正当性を検証するシステムは，図1に示すように，2つのサブシステムに分けて考える事ができる。

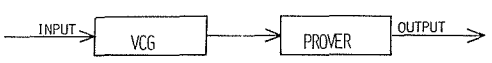

図1　プログラムの検証システム

*　現在は防衛大学校，電気工学教室
**　富士通株式会社
***　情報工学専攻　情報数理工学第一講座

(i) **VCG** (Verification Condition Generator)

このシステムの入力は，入力条件，出力条件及び帰納的アサーションが付加されたプログラムである。入力条件，出力条件及び帰納的アサーションは論理式で与えられる。このシステムの出力は，プログラムの正当性に対する十分条件を表わす論理式で，**検証条件**と言われる。検証条件は，プログラミング言語に適当な演繹体系を導入する事によって，プログラムそれ自身から導びかれる[1]。プログラミング言語 PASCAL[2] に対する VCG が，S. Igarashi, et al.[3] によって試作されている。

(ii) **PROVER** (Theorem Prover)

PROVER は，VCG から出力された検証条件が真であるか否かを決定するシステムである。すなわち，入力は検証条件であり，出力は TRUE 又は UNDECIDABLE である。出力が TRUE ならば，検証条件はすべて真であり，プログラムが正当である事が言える。

本研究では，プログラムの正当性の検証システムを完成させる為に，J. C. King[4] の PROVER をベースとした新たな PROVER を開発した。この事によって，実際的なプログラムの自動検証が可能となった。

2. PROVER

前節で触れた King の PROVER は，加法的クラス*の論理式に対する処理能力を持つ。このクラスに対しては，完全な PROVER が出来る事が知られている[5]。ところで，“一般的には，帰納的アサーションは，加法的クラスの論理式では表現し得ない”という事が知られている。従って帰納的アサーションを表現する為には，適当な関数，述語等を新たに用意する必要がある。しかし，この関数，述語がそのまま検証条件に反映する為，そのままでは検証条件の電子計算機による証明が不可能となる。これを可能とする為には，何らかの形で，新たに用意した関数，述語のデータを与えてやる必要がある。以上の問題点を考慮し，我々は以下に示す方針に沿って PROVER を試作した。

1. 検証条件を表現する為に，適当な関数を用意する。そして，この関数の持つ性質を適当な論理式で表現する。この論理式をトートロジーと解釈し，データとして入力する。
2. PROVER はトートロジーを公理系とする演繹体系を持つものとする。すなわち，論理式の証明は，トートロジーの除去という簡単化によって行なわれる。トートロジーは，予め与えられたものと，1で述べた，関数を処理する為に随時入力されるトートロジーよりなる。

我々が試作した PROVER は，図2に示すように6つのサブシステムよりなるシステムとした。このシステムが処理する論理式は以下に示す記号よりなる論理式とする。

関数名：1字以上の英字の列
変数：1字の英字
定数：整数
論理記号：TRUE, FALSE, ⊃**, &***, ¬****
その他：等号，不等号，＋，－

---

* 自然数上で定義され，定数（整数），変数，加法，減法，整数との積のみから構成される算術式よりなる論理式。
** “ならば”という意味。計算機内では IMPLIES で表現
*** 論理積記号
**** 否定記号

次に各サブシステムの機能を示す。

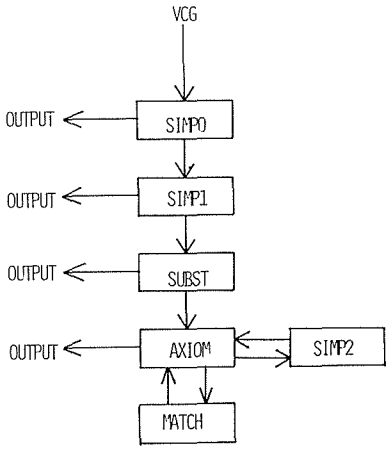

図 2　PROVER の概略

SIMP 0 の機能

このサブシステムは，算術式を一定の形に変形する機能を持つ。変形は，以下に示す順序で行なわれる。

1. 括弧をはずして，式を展開する。
2. 各項の変数は，アルファベット順に並び変えられる。
3. 同類項はまとめられ，式中の各項は辞書式順序で並び変えられる。
4. 関数の引き数も，1, 2, 3, の手続きによって変形される。
5. 不等号は，すべて“＞,”“＜”に変えられる。
6. 定数項がない場合は，“＋0”を付け加える。
7. 係数がない項は，“1＊”を付け加える。

SIMP 1 の機能

このサブシステムは，以下のトートロジーによって論理式の簡単化を行う。

T 1.　$(A \supset B) \supset (A \& B \equiv A \& \mathrm{TRUE})$

T 2.　$(A \& B \equiv C) \supset (A \& B \equiv \mathrm{TRUE} \& C)$

これらのトートロジーは，$P \& A \& B$ という形の論理式に適用される。

ただし，A: $E - C_1\{R_1\}0$*

B: $E - C_2\{R_2\}0$

E: 算術式

C_i: 定数　$(i=1, 2)$

R_i: (不)等号　$(i=1, 2)$

この論理式は，表1の条件を満足する時，$P \& A \& \mathrm{TRUE}$ に変形される。これは，トートロジー T1 による。

また，この論理式は，表2の条件を満足する時，$P \& \mathrm{TRUE} \& C$ に変形される。これは，トートロジー T2 による。

ただし，C: $E - C_3 = 0$, C_3 の値は表2による。

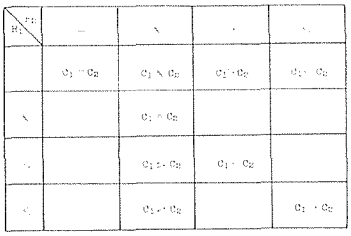
表 1

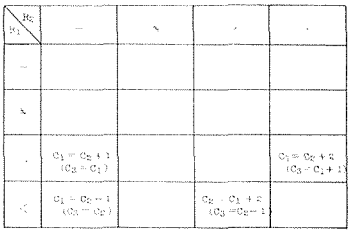
表 2

SIMP 2 の機能

このサブシステムは，以下のトートロジーによって，論理式の簡単化を行う。

---

* 例えば，E: $2 \ast X + 3 \ast Y$, C_1: 4, R_1: $\geqq$ の時
A: $2 \ast X + 3 \ast Y - 4 \geqq 0$ という意味である。

T3. $(A\supset B)\supset((A\&X\supset B\&Y)\equiv(A\&X\supset \mathrm{TRUE}\&Y))$

T4. $(A\&B\equiv C)\supset((A\&B\&X\supset C\&Y)\equiv(A\&B\&X\supset \mathrm{TRUE}\&Y))$

T3 の適用範囲は，T1 と同様である。

T4 は次の様な形の論理式に適用される。

$$X\&A\&B\supset C\&Y$$

ただし，A: $E_1+E_0\{R_1\}0$

B: $E_2-E_0\{R_1\}0$

C: $E_1+E_2\{R_1\}0$

R_1: (不) 等号

E_0: 項

E_1, E_2: 式

SUBST の機能

このサブシステムは，以下のトートロジーによって，論理式の簡単化を行う。

T5. $(P(X)\&X=e\supset Q(X))\equiv(P(e)\&\mathrm{TRUE}\supset Q(e))$

ここで，$P(X)$, $Q(X)$ は，変数 X を含む論理式，e は式である。

AXIOM, MATCH の機能

AXIOM は，新たに定義された関数の処理を行うシステムである。このシステムは，MATCH, SIMP 2 のサポートを必要とする。

まず，次の定理を導入する。

定理 $P\supset f_1=f_2$ かつ $A\supset P$ ならば，$A\supset B\equiv A|^{f_1}_{f_2}\supset B|^{f_1}_{f_2}$

ここで，f_1, f_2 は式，$A|^{f_1}_{f_2}$ は論理式 A 中に現われる f_1 を f_2 で置き換えた論理式を示すものとする。

$P\supset f_1=f_2$ は，与えられた関数の性質を表わす論理式である。この論理式をトートロジーと解釈して，論理式の簡単化を行う。この際，$A\supset P$ のチェックを MATCH 又は SIMP 2 で行なう。MATCH は論理式 P に現われる変数を，A に現われる式と対応させる事によって，$A=A'\&P|^{X_1,\cdots,X_n}_{e_1,\cdots,e_n}$ であるか否かをチェックする。$A=A'\&P|^{X_1,\cdots,X_n}_{e_1,\cdots,e_n}$ ならば，論理式 $P\supset f_1=f_2$ は $(P\supset f_1=f_2)|^{X_1,\cdots,X_n}_{e_1,\cdots,e_n}$ に変換される。この論理式を $P'\supset f_1'=f_2'$ で表わしておく。MATCH でチェックができない場合は，SIMP 2 でチェックされる。

$A=A'\&P'$ が証明された段階で，$A\supset B$ を，前述の定理によって $A|^{f_1'}_{f_2'}\supset B|^{f_1'}_{f_2'}$ に置き換える処理が AXIOM によって行なわれる。

3. 検 証 例

例

```
ENTRY N>=1;
BEGIN
I:=1;
ASSERT N>=I-1 & E(S, X, 1, I)=1 & N>=1;
WHILE I<=N DO
    S(I):=X(I);
```

```
    I:=I+1
    END;
A1: ASSERT G(X, 1, N)=G(S, 1, N) & 1<=N;
D:=S(1);
C:=0;
I:=2;
J:=1;
ASSERT G(X, 1, N)=G(S, 1, N) & B(S, 1, I-1)=S(J) & S(J)=D &
        BB(S, 1, I-1, J)=C & I-1<=N & B(S, 1, I-1)=D;
WHILE I<=N DO
    IF S(I)>D THEN
        BEGIN
        C:=D;
        D:=S(J);
        J:=I
        END;
    ELSE
        BEGIN
        IF S(I)<=C THEN
            C:=S(I)
            END;
        END;
IF C]=0 THEN
    BEGIN
    S(J):=D-C;
    GO TO A1
    END;
Z:=D
END;
EXIT Z=G(X, 1, N);
```

ここで，ENTRY 文，ASSERT 文，EXIT 文は，それぞれ，入力条件，出力条件，帰納的アサーションを与える命令文である。

このプログラムは，配列 X(I) (I=1, N) の最大公約数を求め，その値を Z とするプログラムである。このプログラムの入出力条件及び，帰納的アサーションを与える為に，4つの関数が新たに導入されている。これらの関数の定義を以下に示す。

$$B(S, I, J)=\max\{S(I), S(I+1), \ldots, S(J-1), S(J)\}$$

$$BB(S, I, J, K)=\max\{S(I), \ldots, S(K-1), S(K+1), \ldots, S(J)\}$$

$$E(S, X, I, J)=\begin{cases}1 & \text{if } S(K)=X(K) \text{ for } I\leq K\leq J\\ 0 & \text{その他}\end{cases}$$

$$G(X, J, N): X(I) \quad (I=J, \ldots, N) \text{ の最大公約数}$$

これらの関数の性質は，12個の論理式によって表わす事ができる。これを表3に示す。また，

```
A1.  E(X,Y,1,0)=1

A2.  E(X,Y,1,U-1)=1 IMPLIES E(V(X,U,Y(U)),Y,1,U-1)=E(X,Y,1,U)

A3.  F(X,Y,1,U)=1 IMPLIES G(Y,1,U)=G(X,1,U)

A4.  B(X,1,1)=X(1)

A5.  BB(X,1,1,1)=0

A6.  BB(X,1,Y,U)=0 & B(X,1,Y)=X(U) IMPLIES X(U)=G(X,1,Y)

A7.  X(Y)<=X(U) & B(X,1,Y-1)=X(U) IMPLIES B(X,1,Y)=X(U)

A8.  X(Y)<=X(U) & X(Y)>BB(X,1,Y-1,U) IMPLIES BB(X,1,Y,U)=X(Y)

A9.  X(Y)<=BB(X,1,Y-1,U) IMPLIES BB(X,1,Y,U)=BB(X,1,Y-1,U)

A10. X(Y)<=B(X,1,Y-1) & B(X,1,Y-1)=X(U) IMPLIES B(X,1,Y)=X(U)

A11. X(Y)>B(X,1,Y-1) IMPLIES B(X,1,Y)=X(Y)

A12. X(Y)>B(X,1,Y-1) & B(X,1,Y-1)=X(U) IMPLIES BB(X,1,Y,Y)=X(U)
```

表 3

```
V1.  N>=1
     IMPLIES
     N>=1 & N>=1-1 & E(S,X,1,1-1)=1

V2.  N>=1 & N>=I-1 & E(S,X,1,I-1)=1 & I<=N
     IMPLIES
     N>=1 & N>=(I+1)-1 & F(V(S,I,X(I)),X,1,1)=1

V3.  N>=1 & N>=I-1 & E(S,X,1,I-1)=1 & ¬(I<=N)
     IMPLIES
     G(X,1,N)=G(S,1,N) & N>=1

V4.  G(X,1,N)=G(S,1,N) & N>=1
     IMPLIES
     G(X,1,N)=G(S,1,N) & B(S,1,1)=S(1) & S(1)=(S(1)) & BB(S,1,1,1)=0 & 2-1
     <=N & B(S,1,2-1)=(S(1))

V5.  G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & BB(S,1,I-1,J)=C & I-1<
     =N & B(S,1,I-1)=D & 1<=N
     IMPLIES
     G(X,1,N)=G(V(S,J,(D-C)),1,N) & 1<=N

V6.  G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & BB(S,1,I-1,J)=C & I-1<
     =N & 1<=N & BB(S,1,I-1,J)=C & ¬(I<=N) & C=0
     IMPLIES
     D=G(X,1,N)

V7.  G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & BB(S,1,I-1,J)=C & I-1<
     =N & B(S,1,I-1)=D & I<=N & ¬(S(I)>D) & S(I)>C
     IMPLIES
     G(X,1,N)=G(S,1,N) & B(S,1,(I+1)-1)=S(J) & S(J)=D & BB(S,1,(I+1)-1,J)=
     (S(J)) & (I+1)-1<=N & B(S,1,(I+1)-1)=D & 1<=N

V8.  G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & BB(S,1,I-1,J)=C & I-1<
     =N & B(S,1,I-1)=D & 1<=N & I<=N & ¬(S(I)>D) & ¬(S(I)<C)
     IMPLIES
     G(X,1,N)=G(S,1,N) & B(S,1,(I+1)-1)=S(J) & S(J)=D & BB(S,1,(I+1)-1,J)=
     C & (I+1)-1<=N & B(S,1,(I+1)-1)=D

V9.  G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & BB(S,1,I-1,J)=C & I-2<
     =N & B(S,1,I-1)=D & 1<=N & I<=N & S(I)>D
     IMPLIES
     G(X,1,N)=G(S,1,N) & B(S,1,(I+1)-1)=S(I) & S(I)=(S(I)) & B(S,1,(I+1)-1
     ,I)=D & (I+1)-1<=N & B(S,1,(I+1)-1)=D
```

表 4

VCG から出力される検証条件は 9 個である。これを表 4 に示す。ここで V(S, I, Y) という形の記号は，配列 S の I 番目の要素が Y に置き換えられた配列を意味する。

検証条件 V6 がどのようにして証明されるかを，図 3 に示す。他の検証条件も同様な過程を経て証明されているが，ここでは省略する。

```
THE VERIFICATION CONDITION IS FOLLOWING.

G(X,1,N)=G(S,1,N) & B(S,1,I-1)=S(J) & S(J)=D & I-1<=N & 1<=N & BB(S,1,
I-1,J)=C & ¬(I<=N) & C=0
IMPLIES
D=G(X,1,N)


THE V.C. IS TRANSFORMED TO THE FOLLOWING FORM.

1*G(X,1,N)-1*G(S,1,N)+0 =0 & 1*B(S,1,I-1)-1*S(J)+0 =0 & 1*S(J)-1*D+0 =
0 & 1*N-1*I+2 >0 & 1*N+0 >0 & 1*BB(S,1,I-1,J)-1*C+0 =0 & 1*N-1*I+0 <0
& 1*C+0 =0
IMPLIES
1*G(X,1,N)-1*D+0 =0


CALL SUBPROGRAM SIMP1

THE V.C. IS TRANSFORMED TO THE FOLLOWING FORM.

1*G(X,1,N)-1*G(S,1,N)+0 =0 & 1*B(S,1,I-1)-1*S(J)+0 =0 & 1*S(J)-1*D+0 =
0 & TRUE & 1*N+0 >0 & 1*BB(S,1,I-1,J)-1*C+0 =0 & 1*N-1*I+1 =0 & 1*C+0
=0
IMPLIES
1*G(X,1,N)-1*D+0 =0


CALL SUBPROGRAM SUBST

THE V.C. IS TRANSFORMED TO THE FOLLOWING FORM.

1*G(X,1,I-1)-1*G(S,1,I-1)+0 =0 & 1*B(S,1,I-1)-1*S(J)+0 =0 & TRUE & TRU
E & 1*I-1 >0 & TRUE & TRUE & 1*BB(S,1,I-1,J)+0 =0
IMPLIES
1*G(X,1,I-1)-1*S(J)+0 =0


CALL SUBPROGRAM AXIOM
CALL SUBPROGRAM MATCH
CALL SUBPROGRAM MATCH
CALL SUBPROGRAM MATCH
CALL SUBPROGRAM MATCH

THE V.C. IS TRANSFORMED TO THE FOLLOWING FORM.

1*G(X,1,I-1)-1*G(S,1,I-1)+0 =0 & 1*B(S,1,I-1)-1*G(S,1,I-1)+0 =0 & TRUE
 & TRUE & 1*I-1 >0 & TRUE & TRUE & 1*BB(S,1,I-1,J)+0 =0
IMPLIES
1*G(X,1,I-1)-1*G(S,1,I-1)+0 =0


THIS FORMULA IS VALID.
```

図 3 検証条件 V 6 の証明図

1) 検証条件は，SIMP 0 によって標準形に変形される。
2) (N−I>−2 & N−I<0)≡(N−I=1) であるから，SIMP 1 に於て T 2 が適用される。
3) SUBST によって，変数 D, N がそれぞれ S(J), I-1 に置き換えられる。
4) 表の 3 A6 に於いて，U, Y, X をそれぞれ J, I-1, S で置き換えて得られる論理式は，定理の条件をみたす。これは，MATCH によってチェックされる。よって，定理より，S(J) が G(S, 1, I-1) に置き換えられる。この処理は，AXIOM によってなされる。
5) 最終的に得られた論理式は，$A\&B\supset A$ という形で，この論理式は恒真式であるから証明が完了した。

4. あ と が き

今回試作した PROVER は，加法的クラスの論理式に対しては，完全でない。これは，予め与えてあるトートロジーの集合が不十分な為である。しかしながら，実際のプログラム検証に於ては，それ程重要な問題点とはならない。これは，検証条件が，大体同じパターンの論理式である事による。以前にも述べた通り，完全な PROVER を作る事は可能であるが，処理時間，記憶容量の面から見ると，実際的とは言えない。

今回試作したシステムで検証したプログラムは，最大で約 30 ステップ程度の比較的小さなプログラムだけであった。更に大きなプログラムに対しては，帰納的アサーションの割り当てという難しい問題がある。しかしながら，プログラムをブロックに分けて細分化する事によって，よ

り大きなプログラムの検証も可能であろう。

参 考 文 献

1) 林，五十嵐：情報処理学会誌，17 (1976), p. 437-447.
2) Wirth, N.: Acta Informasica, 1 (1971), p. 35-63.
3) Igarashi, S., London, R. L. and Luckham, D. C.: Acta Informatica, 4 (1975), p. 145-182.
4) King, J. C.: JCSS, 6 (1972), p. 305-323.
5) Cooper, D. C.: Machine Intelligence, 6 (1971), p. 43-59.