# 用 Hoare 邏輯驗證程序的一般方法及實例*

楊 靜

（貴州大學計算機科學與技術學院，貴陽 550025）

**摘 要**：本文引用一個簡單的語言說明如何驗證一個程序的正確性，并且給出一個實例來進行驗證，并指明了今後的研究方向。

**關鍵詞**：Hoare 邏輯；部分正確性；公理化系統

## 1. 引 言

Hoare 邏輯[1-2]提供了一個驗證程序正確性的基礎。程序的部分正確性可表示爲{P}C{Q}，稱爲 Hoare 三元組或斷言，其中 P 和 Q 爲一階邏輯公式，分別表示前置條件和後置條件，C 爲程序。{P}C{Q} 的意思爲：如果 P 在執行 C 前爲真且 C 能够終止，那麽 Q 在 C 終止時爲真。Hoare 邏輯包括一些公理和規則。這些公理和規則與程序中使用的命令密切相關。

## 2. 方法描述

本文用一個簡單的 while 語句來説明如何驗證程序的正確性。while 語言中有以下命令：

S::= x:= a |skip| $S_1$; $S_2$ | if b then $S_1$ else $S_2$ | while b do S

其中 $x$ 爲變量名，$a$ 爲算術表達式，$b$ 爲布爾表達式。

采用如表 1 所示的公理化系統。

公理[ass]中，謂詞 P[x↦a]表示把 P 中出現的 $x$ 替換成 $a$ 得到的謂詞。這條公理表示如果執行賦值語句 x:=a 後 P 爲真，那麽在執行賦值語句之前 P[x↦a]就應該爲真，注意 $x$ 是自由變元。

公理[skip]表示如果 P 在 skip 語句執行之前爲真，那麽 skip 語句的執行不會改變 P 的真值。

規則[comp]表示 {P}$S_1$; $S_2${R} 的成立需要以 {P}$S_1${Q}和{Q}$S_2${R}的成立爲前提條件，注意 Q 起着一個橋梁的作用。

表 1 公理化系統

| 規則名稱 | 規則内容 |
|---|---|
| [ass] | {P[x↦a]} x: = a{P} |
| [skip] | {P} skip {P} |
| [comp] | $\frac{\{P\}S_1\{Q\},\{Q\}S_2\{R\}}{\{P\}S_1;S_2\{R\}}$ |
| [if] | $\frac{\{b\wedge P\}S_1\{Q\},\{\neg b\wedge P\}S_2\{Q\}}{\{P\}\text{if } b \text{ then } S_1 \text{ else } S_2\{Q\}}$ |
| [while] | $\frac{\{b\wedge P\}S\{P\}}{\{P\}\text{while } b \text{ do } S\{\neg b\wedge P\}}$ |
| [cons] | $\frac{\{P'\}S\{Q'\}}{\{P\}S\{Q\}}$ 如果 P⇒P′, Q′⇒Q |

規則[if]中的兩個前提表示如果執行 $S_1$，那麽在執行前 P 和 b 都應該爲真；如果執行 $S_2$，那麽在執行前 P 爲真而 b 爲假。

規則[while]中出現的謂詞 P，其實是循環中必須保持爲真的不變式[3]。

規則[cons]是一條表示因果關系的規則，它表示如果一個 Hoare 三元組爲真，那麽把前置條件加强或後置條件減弱也應該爲真。

爲了符合邏輯學的習慣，以下用⊢{P}S{Q}表示斷言{P}S{Q}的可證性。

## 3. 實 例

下面用一個簡單的程序來説明如何推證一個程序的部分正確性，實際上就是推證一個程序是否

* 本文得到貴州大學引進人才科研基金資助項目和貴州省省長基金資助項目（No. 2005（212））的資助。

【作者簡介】楊靜（1965－），副教授，博士；研究方向：軟件理論及形式化方法。

滿足客户的需求。

考慮關于程序 i:=0; s:=0; while¬(i=n) do (i:=i+1; s:=s+i)的部分正確性，其中 $n$ 是一個給定的自然數。

第一步：先證明

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1;\ s:=s+i\ \{i>0\wedge s=\sum_{j=1}^{i} j\}$$

證明過程如下：

$$\vdash \{i>0\wedge s+i=\sum_{j=1}^{i} j\}\ s:=s+i\ \{i>0\wedge s=\sum_{j=1}^{i} j\}\quad [ass]$$

因爲 $i>0\wedge s=\sum_{j=1}^{i-1} j \Rightarrow i>0\wedge s+i=\sum_{j=1}^{i} j$，所以

$$\vdash \{i>0\wedge s=\sum_{j=1}^{i-1} j\}\ s:=s+i\ \{i>0\wedge s=\sum_{j=1}^{i} j\}\quad [cons]\quad (1)$$

又 $\vdash \{i+1>0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1\ \{i>0\wedge s=\sum_{j=1}^{i-1} j\}$ [ass]

及 $i\geq 0\wedge s=\sum_{j=1}^{i} j \Rightarrow i+1>0\wedge s=\sum_{j=1}^{i} j$ 得：

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1\ \{i>0\wedge s=\sum_{j=1}^{i-1} j\}\quad [cons]\quad (2)$$

由式(1)及式(2)得：

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1;\ s:=s+i\ \{i>0\wedge s=\sum_{j=1}^{i} j\}\quad (3)$$

第二步：進一步尋找循環的不變式。

由式(3)及 $i>0\wedge s=\sum_{j=1}^{i} j \Rightarrow i\geq 0\wedge s=\sum_{j=1}^{i} j$ 得：

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1;\ s:=s+i\ \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ [cons]\quad (4)$$

所以循環的不變式爲 $i\geq 0\wedge s=\sum_{j=1}^{i} j$，而 $\neg(i=n)\wedge i\geq 0\wedge s=\sum_{j=1}^{i} j \Rightarrow i\geq 0\wedge s=\sum_{j=1}^{i} j$，從而

$$\vdash \{\neg(i=n)\wedge i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ i:=i+1;\ s:=s+i\ \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\quad [cons]\quad (5)$$

第三步：最終證明原程序的部分正確性。

利用式(5)得：

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ \text{while}\neg(i=n)\ \text{do}\ (i:=i+1;\ s:=s+i)\ \{i=n\wedge i\geq 0\wedge s=\sum_{j=1}^{i} j\}\quad [while]\quad (6)$$

由 $i=n\wedge i\geq 0\wedge s=\sum_{j=1}^{i} j \Rightarrow i=n\wedge s=\sum_{j=1}^{i} j$ 及式(6)得：

$$\vdash \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\ \text{while}\ \neg(i=n)\text{do}\ (i:=i+1;\ s:=s+i)\ \{i=n\wedge s=\sum_{j=1}^{i} j\}\quad [cons]\quad (7)$$

使用[ass]兩次可得：

$$\vdash \{i\geq 0\wedge 0=\sum_{j=1}^{i} j\}\ s:=0\ \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\quad [ass]\quad (8)$$

$\vdash \{0\geq 0\wedge 0=\sum_{j=1}^{0} j\}\ i:=0\ \{i\geq 0\wedge 0=\sum_{j=1}^{i} j\}$，其前置條件爲 true，即：

$$\vdash \{true\}\ i:=0\ \{i\geq 0\wedge 0=\sum_{j=1}^{i} j\}\quad [ass]\quad (9)$$

聯合式(8)及式(9)可得：

$$\vdash \{true\}\ i:=0;\ s:=0\ \{i\geq 0\wedge s=\sum_{j=1}^{i} j\}\quad [comp]\quad (10)$$

利用式(7)和式(10)得：

$$\vdash \{true\}\ i:=0;\ s:=0;\ \text{while}\neg(i=n)\ \text{do}\ (i:=i+1;\ s:=s+i)\ \{i=n\wedge s=\sum_{j=1}^{i} j\}\quad [comp]$$

這就是原程序段關于部分正確性的斷言，它説明無論從任何一個狀態出發，衹要這個程序能够終止，那麽 S 就是從 1 到 n 的和。

## 4. 結　論

這種方法可以推廣到驗證面向對象程序的正確性，基本思想是用 Hoare 邏輯的推廣——分離邏輯[4]來完成。這方面的工作是非常前沿性的工作，

值得進一步探討。

**参考文献:**

[1] C.A.R Hoare. An Axiomatic Basis for Computer Programming. *Communications of ACM*, 1969(12): 576-583.

[2] Hongseok Yang. Hoare Logic. Electronic Manuscript, July 2003.

[3] David Gries. *The Science of Programming*. Springer, 1981.

[4] John Creynolds. An Introduction to Separation Logic. Class Notes for CS 819A4, Springer, 2003.

# General Way and Example to Verify the Correctness of Program Using Hoare Logic

*YANG Jing*

**Abstract:** This paper shows how to verify the correctness of program with respect to a language in the frame of Hoare logic, and gives a detailed example to validate this method. The future research direction is also proposed.

**Key words:** Hoare logic; partial correctness; axiomatic system

（責任編輯：Martina, Susan）

（上接第 41 頁）

# Multivectors-based Scheme for Congestion Control in Wireless Sensor Networks

*CHEN Hai-guang, HAN Peng, WU Hua-feng, GAO Chuan-shan*

**Abstract:** Wireless sensor networks have wide applications fields such as military and intelligent home management due to ease deployment of these sensor nodes. It is important to build an efficient, fair, power-efficient transport scheme to gather the message and send message to the sink or base-station. This paper proposes a distributed, dynamic, based-on route path multi-vector control mechanism. It can quickly detect congestion, and send the feedback signal to the downstream node. The scheme includes the traffic of different nodes, the total number of sub-nodes and the levels of the sub-tree. Simulation results show that the proposed management scheme can achieve a good level of fairness, power-efficient and stability.

**Key words:** wireless sensor networks; distributed; fairness; congestion control; power efficient

（責任編輯：Susan, Becky, Martina）