NJUSWI
2002-2012
Anniversary
南京大學 软件学院

顺序、循环、分支

刘 钦

南京大学软件学院

结构化程序设计起源

- 科拉多·伯姆及朱塞佩·贾可皮尼（Giuseppe Jacopini）
  - 1966年5月
  - 《Communications of the ACM》（Flow diagrams, turing machines and languages with only two formation rules）期刊发表论文
  - 说明任何一个有goto指令的程序，可以改为完全不使用goto指令的程序
- 艾兹赫尔·戴克斯特拉（Edsger Wybe Dijkstra）
  - 1968年
  - 《GOTO陈述有害论》（Go To Statement Considered Harmful）
  - 因此结构化程序设计开始盛行，此概念理论上可以由结构化程序理论所证明

底层的结构化程序设计

- 结构化的程序是以一些简单、有层次的程序流程架构所组成，可分为循序（sequence）、选择（selection）及重复（repetition）。
  - 循序是指程序正常的运行方式，运行完一个指令后，运行后面的指令。
  - 选择是依程序的状态，选择数段程序中的一个来运行，一般会使用if..then..else..endif或switch、case等关系字来识别。
  - 重复是指一直运行某一段程序，直到满足特定条件，或是一集合体中的所有元素均已处理过，一般会使用while、repeat、for或do..until等关键字识别。一般会建议每个循环只能有一个进入点（戴克斯特拉的结构化程序设计要求每个循环只能有一个进入点及一个结束点，有些编程语言仍有此规定）。

如何证明程序的正确性呢

- Edsger W. Dijkstra
  - Enumeration
  - Mathematical induction
  - Abstraction
- C. A. R. HOARE
  - Axioms proof

How to proof — Edsger W. Dijkstra

- Two statements
  - int r = a; int dd =d;
  - while (dd<=r)
    - dd=2*dd;
  - while (dd!=d){
    - dd = dd/2;   //halve dd;
    - if（dd<=r）{ r= r-dd;} //reduce r modulo dd
  - }
- operating on the variable “r” and “dd” leaves the relations
  - 0 <= r < dd
    - which is satisfied to start with

It is asked to establish that the successive execution of the following two statements

$$\text{“}dd := dd/2;$$
$$\textbf{if } dd \leqslant r \textbf{ do } r := r - dd\text{”}$$

operating on the variables “r” and “dd” leaves the relations

$$0 \leqslant r < dd \tag{1}$$

invariant. One just “follows” the little piece of program assuming that (1) is satisfied to start with. After the execution of the first statement, which halves the value of dd, but leaves r unchanged, the relations

$$0 \leqslant r < 2*dd \tag{2}$$

will hold. Now we distinguish two mutually exclusive cases.

(1) $dd \leqslant r$. Together with (2) this leads to the relations

$$dd \leqslant r < 2*dd; \tag{3}$$

In this case the statement following **do** will be executed, ordering a decrease of r by dd, so that from (3) it follows that eventually

$$0 \leqslant r < dd,$$

i.e. (1) will be satisfied.

(2) **non** $dd \leqslant r$ (i.e. $dd > r$). In this case the statement following **do** will be skipped and therefore also r has its final value. In this case “$dd > r$” together with (2), which is valid after the execution of the first statement leads immediately to

$$0 \leqslant r < dd$$

so that also in the second case (1) will be satisfied.

Thus we have completed our proof of the invariance of relations (1), we have also completed our example of enumerative reasoning, conditional clauses included.

Enumeration

TABLE I

A1	$x + y = y + x$	addition is commutative
A2	$x \times y = y \times x$	multiplication is commutative
A3	$(x + y) + z = x + (y + z)$	addition is associative
A4	$(x \times y) \times z = x \times (y \times z)$	multiplication is associative
A5	$x \times (y + z) = x \times y + x \times z$	multiplication distributes through addition
A6	$y \leqslant x \supset (x - y) + y = x$	addition cancels subtraction
A7	$x + 0 = x$	
A8	$x \times 0 = 0$	
A9	$x \times 1 = x$	

$$y \leqslant r \supset r + y \times q = (r - y) + y \times (1 + q)$$

The proof of the second of these is:

A5	$(r - y) + y \times (1 + q)$
	$= (r - y) + (y \times 1 + y \times q)$
A9	$= (r - y) + (y + y \times q)$
A3	$= ((r - y) + y) + y \times q$
A6	$= r + y \times q$ provided $y \leqslant r$

How to proof — Hoare

D0 Axiom of Assignment

$\vdash P_0 \{x := f\}\, P$

where

x is a variable identifier;

f is an expression;

P_0 is obtained from P by substituting f for all occurrences of x.

D1 Rules of Consequence

If $\vdash P\{Q\}R$ and $\vdash R \supset S$ then $\vdash P\{Q\}S$

If $\vdash P\{Q\}R$ and $\vdash S \supset P$ then $\vdash S\{Q\}R$

D2 Rule of Composition

If $\vdash P\{Q_1\}R_1$ and $\vdash R_1\{Q_2\}R$ then $\vdash P\{(Q_1\,;\,Q_2)\}R$

D3 Rule of Iteration

If $\vdash P \wedge B\{S\}P$ then $\vdash P\{\mathbf{while}\ B\ \mathbf{do}\ S\}\neg B \wedge P$

TABLE III

Line number	Formal proof	Justification
1	$\mathbf{true} \supset x = x + y \times 0$	Lemma 1
2	$x = x + y \times 0\{r := x\}x = r + y \times 0$	D0
3	$x = r + y \times 0\ \{q := 0\}\ x = r + y \times q$	D0
4	$\mathbf{true}\ \{r := x\}\ x = r + y \times 0$	D1 (1, 2)
5	$\mathbf{true}\ \{r := x;\ \ q := 0\}\ x = r + y \times q$	D2 (4, 3)
6	$x = r + y \times q \wedge y \leqslant r \supset x = (r-y) + y \times (1+q)$	Lemma 2
7	$x = (r-y) + y \times (1+q)\{r := r-y\}x = r + y \times (1+q)$	D0
8	$x = r + y \times (1+q)\{q := 1+q\}x = r + y \times q$	D0
9	$x = (r-y) + y \times (1+q)\{r := r-y;\ q := 1+q\}\ x = r + y \times q$	D2 (7, 8)
10	$x = r + y \times q \wedge y \leqslant r\ \{r := r-y;\ q := 1+q\}\ x = r + y \times q$	D1 (6, 9)
11	$x = r + y \times q\ \{\mathbf{while}\ y \leqslant r\ \mathbf{do}\ (r := r-y;\ \ q := 1+q)\}\ \neg y \leqslant r \wedge x = r + y \times q$	D3 (10)
12	$\mathbf{true}\ \{((r := x;\ \ q := 0);\ \ \mathbf{while}\ y \leqslant r\ \mathbf{do}\ (r := r-y;\ \ q := 1+q))\}\ \neg y \leqslant r \wedge x = r + y \times q$	D2 (5, 11)