TATA Institute Bombay
Jan 1978

THEORY OF PROGRAMMING (1)

and its application to the design of correct and efficient computer programs.

A COURSE OF LECTURES

given by Professor C.A.R. Hoare

Professor of Computation at Oxford University

BASED ON THE TEXT (2)

a
discipline
of
programming

Edsger W. Dijkstra

Prentice Hall, 1976

BASIC DEFINITIONS (3)

A <u>number</u> is denoted by a string of digits

e.g. 0 3 03 703

A <u>variable</u> is denoted by a string of letters

e.g. X Y ALPHA QUOT

A <u>machine state</u> is a finite mapping between variables and values.

e.g. m =

X	Y	REM	QUOT
37	7		0

m(X) = 37, m(Y) = 7, m(QUOT) = 0

m(REM) is undefined. m is also undefined for all other variables.

A command C is a relation between machine states, i.e. the states of the machine before and after the command is obeyed. (4)

(we define a relation as a set of ordered pairs $(m,m') \in C$).

The skip command is defined:

$$skip =_{df} \{(m,m') \mid m = m'\}$$

i.e. the identity relation.

This command is obeyed by doing nothing!

The abort command is defined

$$abort = \{\ \}$$

i.e. the empty relation.

A machine which attempts to obey this command will simply fail (break)!

An expression is constructed from variables, values, operators, and brackets. (5)

e.g. $X \quad 17 \quad X + 17 \quad Y*(X - 3)$

Given machine state m and expression e, we define $m^*(e)$ - as the value taken by e, when evaluated in machine state m; i.e. when its variables are replaced by the values given by m.

e.g. if $m(X) = 3$ then $m^*(X + 17) = 20$

$m^*(17) = 17$

if $m(Y)$ is not defined, then nor is

$m^*(Y*(X + 3))$

An assignment command takes the form (6)

x := e (x becomes y)

where x is a variable

and e is an expression

e.g. Y := 17 Y := X − 1 X := X + 1.

$$x := e =_{df} \{ (m,m') \mid m'(x) = m^*(e). \; \& \forall y \neq x \; m'(y) = m(y) \}$$

It is obeyed by evaluating e in the initial machine state m, and then changing m(x) to have this value instead of its old one. If e is undefined in m, the machine breaks.

The composition of commands c1 and c2 is: (7)

$$c1;\ c2 =_{df} \{ (m,m') \mid \exists m'' \ (m,m'') \in c1 \ \& \ (m'',m') \in c2 \}$$

It is obeyed by first obeying c1 and then obeying c2. m" is the final machine state of c1 and the initial machine state of c2.

Theorem. c1;(c2; c3) = (c1; c2); c3

The associativity of ; will justify omission of brackets.

Theorem. skip; c = c; skip = c

abort; c = c; abort = abort

Compare: 1*c = c*1 = c

0*c = c*0 = 0

A condition b is an expression which is either true $\underline{T}$ or false $\underline{F}$ (8)

e.g. X 0 $\quad X = Y \quad \underline{T} \quad \underline{F}$

It defines a "command"

$$\{ (m,m') \mid m^*(b) = \underline{T} \ \& \ m' = m \}$$

i.e. the identity relation restricted to those machine states in which b is true.

It is obeyed by evaluating b; if this is true, skip ; otherwise abort.

A conditional command is defined

$$\underline{\text{if}}\ b \rightarrow c1,\ c2\ \underline{\text{fi}} = b;\ c1 \cup \bar{b};\ c2$$

It is obeyed by first evaluating b; if the value is true, c1 is obeyed and c2 omitted. if the value is false, c2 is obeyed and c1 omitted.

$$\underline{\text{if}}\ b \rightarrow \text{skip, abort}\ \underline{\text{fi}} = b$$

Theorems $\underline{\text{if}}\ b \rightarrow c1,\ c2\ \underline{\text{fi}}\ ;\ c3 = \underline{\text{if}}\ b \rightarrow (c1;\ c3),\ (c2;\ c3)\ \underline{\text{fi}}$

$$\underline{\text{if}}\ b \rightarrow c1,\ c2\ \underline{\text{fi}} = \underline{\text{if}}\ \bar{b} \rightarrow c2,\ c1\ \underline{\text{fi}}$$

$$\underline{\text{if}}\ \underline{T} \rightarrow c1,\ c2\ \underline{\text{fi}} = \underline{\text{if}}\ \underline{F} \rightarrow c2,\ c1\ \underline{\text{fi}} = c1.$$

The repetitive command is defined (9)

$$\underline{\text{do}}\ b \rightarrow c\ \underline{\text{od}} = \bigcup_{n=0}^{\infty} c_n \qquad (\underline{\text{while}}\ b\ \underline{\text{do}}\ c)$$

where $c_0 = \bar{b}$

$$c_{n+1} = b;\ c;\ c_n\ [\cup\ \bar{b}]$$

It is obeyed by first evaluating b. If this is false, the task is finished. If it is true, c is next obeyed, and then the whole command is repeated.

Theorem. $\underline{\text{do}}\ b \rightarrow c\ \underline{\text{od}} = \underline{\text{if}}\ b \rightarrow (c;\ \underline{\text{do}}\ b \rightarrow c\ \underline{\text{od}}),\ \text{skip}\ \underline{\text{fi}}$

$$\underline{\text{do}}\ \underline{F} \rightarrow c\ \underline{\text{od}} = \text{skip}$$

$$\underline{\text{do}}\ \underline{T} \rightarrow c\ \underline{\text{od}} = \text{abort}$$

$$\underline{\text{do}}\ b \rightarrow c\ \underline{\text{od}} = \bar{b},\ b;c;\bar{b},\ b;c;b;c;\bar{b},\ \ldots$$

$$c_n \subseteq c_{n+1}$$

$$c_n = \underbrace{b;c;\ b;c\ \ldots\ b;c;}_{\leq n \text{ times.}}\bar{b}$$

EXAMPLE

(10)

$x := X; \quad y := Y;$

$\underline{do}\ x \neq y \rightarrow \underline{if}\ \ x < y \rightarrow\ \ y := y - x,\ x := x - y$

$\underline{fi}$

$\underline{od}$

$= x := X;\ y := Y;$

$\{x = y \cup (x \neq y;\ x < y;\ y := y - x;\ x = y)$

$\cup\ (x \neq y;\ \overline{x < y};\ x := x - y;\ x = y)$

....

$\cup\ x \neq y;\ x < y;\ y := y - x;\ x \neq y;\ \overline{x < y};\ x := x - y;\ x = y$

....

$\}$

EXECUTION TRACES (11)

	X	Y	x	y
initial m/s	111	259		
$x := X$	111	259	111	
$y := Y$	"	"	"	259
$x \neq y$ ✓				
$x < y$				
$y := y - x$				148
$x \neq y;\ x < y$ ✓				
$y := y - x$				37
$x \neq y;\ \overline{x < y}$ ✓				
$x := x - y$			74	
$x \neq y;\ \overline{x < y}$ ✓				
$x := x - y$			37	
$x = y$ ✓				
final m/s.	111	259	37	37

A. Obey the following commands, giving their execution traces. (12)

1. $y := 0;\ x := 3;\ \underline{do}\ y + 1 < x \rightarrow y := y + 1\ \underline{od}$

2. $y := 2;\ x := 3;\ \underline{do}\ y > 0 \rightarrow x := x + 1;\ y := y - 1\ \underline{od}$

3. $y := 2;\ x := 3;\ s := 0;$

   $\underline{do}\ y > 0 \rightarrow y := y - 1;\ s := s + x\ \underline{od}$

4. $y := 3;\ x := 7;\ q := 0;$

   $\underline{do}\ x > y \rightarrow x := x - y;\ q := q + 1\ \underline{od}$

B. In each example above, replace the repetitive command by a pair of assignments which would have the same effect for <u>any</u> initial assignment to x, y, q, s.

e.g. the answer to example 4 is $q := q + x \div y;\ x := x \text{ modulo } y$

WEAKEST PRECONDITIONS (13)

c <u>achieves</u> r is defined:

$$c\ \underline{a}\ r = \{(m,m)\ |\ \exists m'\ (m,m') \in c\ \&\ m'^*(r) = \underline{T}\}$$

i.e. the inverse image of r under c. It is the condition satisfied by exactly those initial machine states in which c can successfully be obeyed, and can end in a machine state satisfying r.

Theorem $b\ \underline{a}\ r = b \cap r$ $\quad\quad b\ \&\ r$

where b is a condition.

Proof. $\text{LHS} = \{mm\ |\ \exists m'\ (m,m') \in b\ \&\ m'^*(r) = \underline{T}\}$

$= \{mm\ |\ (m,m) \in b\ \&\ m^*(r) = \underline{T}\}$

$= \{mm\ |\ (m,m) \in B\ \&\ (m,m) \in r\}$

$= b \cap r$

We usually identify a condition with the set of machine state pairs in which the condition is true.

$$\underline{T}\ \underline{a}\ r = r$$

Strict $\quad c\ \underline{a}\ \underline{F} = \underline{F}, \quad \underline{F}\ \underline{a}\ r = \underline{F}$

Distributive $\quad c\ \underline{a}\ (\bigcup_n r_n) = \bigcup_n (c\ \underline{a}\ r_n)$

(Additive) $\quad (\bigcup_n c_n)\ \underline{a}\ r = \bigcup_n (c_n\ \underline{a}\ r)$

A command c is <u>deterministic</u> if for each initial state m, there is at most <u>one</u> final state m' such that $(m,m') \in c$. All commands defined so far are deterministic.

If c is deterministic.

Multiplicative $\quad c\ \underline{a}(\bigcap_n r_n) = \bigcap_n (c\ \underline{a}\ r_n)$

This is <u>not</u> true if c is nondeterministic,

e.g. $c = x := 0 \bigcup x := 1$

then $\quad c\ \underline{a}\ x = 0 = \underline{T}\ \ \&\ \underline{a}\ \ x = 1 = \underline{T}$

$\quad c\ \underline{a}\ (x = 0\ \&\ x = 1) = c\ \underline{a}\ \underline{F} = \underline{F}$

Theorem of assignment (15)

$$x := e \ \underline{a}\ b(x) \ = b(e)$$

where b(e) is the result of replacing all occurrences of x in b(x) by e.

e.g. $(X := 37 \ \underline{a}\ X > 12) \equiv 37 > 12 \equiv \underline{T}$

$(X := Y*3 \ \underline{a}\ X > 12) \equiv Y*3 > 12 \equiv Y > 4$

$(X := X - 1 \ \underline{a}\ X > 12) \equiv (X - 1 > 12) \equiv (X > 13)$

Proof. The value of x after the assignment is by definition equal to the value of e before the assignment. So b(x) is true (of x) after the assignment if any only if b(e) is true (of the value of e) before the assignment. The values of all other variables of b remain unchanged by the assignment.

Theorem of composition (16)

$$(c1;\ c2)\underline{a}r \ = c1 \ \underline{a}\ (c2 \ \underline{a}\ r)$$

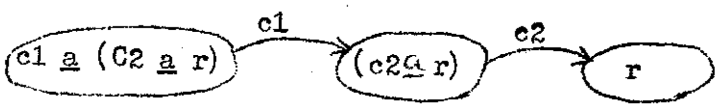
$$(x := X;\ y := Y) \ \underline{a}\ (GCD(x,y) = GCD(X,Y))$$

$$\equiv \quad (x := X \ \underline{a}\ GCD(x,y) = GCD(X,Y))$$

$$\equiv \quad GCD(X,Y) = GCD(X,Y)$$

$$\equiv \quad X > 0 \ \& \ Y > 0$$

Proof. c1; c2 arrives at a state satisfying r if c1 arrives at a state from which c2 achieves r, and conversely



$$\text{LHS} = \{mm \mid \exists m' (m,m') \in (c1;\ c2) \ \& r^*(m') = \underline{T}\}$$

$$= \{mm \mid \exists m' \exists m'' \ (m,m'') \in c1 \ \& (m'',m') \in c2 \ \& r^*(m') = \underline{T}\}$$

$$= \{mm \mid \exists m'' (m,m'') \in c1 \ \& [\exists m' (m'',m') \in c2 \ \& \ r^*(m') = \underline{T}\}$$

$$= \{mm \mid \exists m'' (m,m'') \in c1 \ \& \ m'' \in (c2 \ \underline{a}\ r)\} = \text{RHS}$$

Theorem (16)

$\underline{\text{if}}\ b \rightarrow c1,\ c2\ \underline{\text{fi}}\ \underline{a}\ r = \underline{\text{if}}\ b \rightarrow (c1\ \underline{a}\ r),(c2\ \underline{a}\ r)\ \underline{\text{fi}}$

$= b \cap (c1\ \underline{a}\ r) \cup \overline{b} \cap (c2\ \underline{a}\ r)$

e.g. $(\underline{\text{if}}\ x < y \rightarrow y := y - x,\ x := x - y\ \underline{\text{fi}}\ \underline{a}\ GCD(x,y) = K) \equiv$

$\underline{\text{if}}\ x < y \rightarrow GCD(x, y - x) = K,\ GCD(x - y,\ y) = K\ \underline{\text{fi}}$

$GCD(x,y) = K\ \&\ x \neq y$

Proof. $LHS = (b;\ c1 \cup \overline{b};\ c2)\ \underline{a}\ r$

$= (b;\ c1)\underline{a}\ r \cup (\overline{b};\ c2)\ \underline{a}\ r$

$=\ b\ \underline{a}\ (c1\ \underline{a}\ r) \cup \overline{b}\ \underline{a}(c2\ \underline{a}\ r)$

$=\ b \cap (c1\ \underline{a}\ r) \cup \overline{b} \cap (c2\ \underline{a}\ r)$

$=$ RHS.

Theorem of repetition. (17)

$$\underline{\text{do}}\ b \rightarrow c\ \underline{\text{od}}\ \underline{a}\ r = \bigcup_{n=0}^{\infty} Pn$$

where $Po = \overline{b} \cap r$

$Pn{+}1 = b \cap c\underline{a}\ Pn \cup \overline{b} \cap r$

e.g. $\underline{\text{do}}\ y{+}1 < x \rightarrow y := y{+}1\ \underline{\text{od}}\ \underline{a}\ y = x - 1 = \bigcup_{n=0}^{\infty} Pn$

where $Po \equiv \overline{y{+}1 < x} \cap\ y = x - 1 \equiv y = x - 1$

$P1 \equiv y{+}1 < x \cap (y := y{+}1\ \underline{a}\ Po) \cup\ y = x - 1$

$\equiv y = x - 2 \vee\ y = x - 1$

$\equiv x - 1 - 1 \leqslant y < x$

...

$Pn \equiv x - n - 1 \leqslant y < x$

$$\therefore \bigcup_{n=0}^{\infty} Pn \equiv \exists n\quad x - n - 1 \leqslant y < x$$

$\equiv y < x$

$LHS = \bigcup_n c_n\ \underline{a}\ r$ where $c_o = \overline{b},\quad c_{n+1} = b;c;c_n \cup \overline{b}$

$= \bigcup_n (c_n\ \underline{a}\ r)$ where $c_o\ \underline{a}r = \overline{b} \cap r,\ c_{n+1}\ \underline{a}\ r = (b;c;c_n\underline{a}\ r) \cup \overline{b} \cap r$

$= b \cap (c\ \underline{a}(c_n\ \underline{a}\ r)) \cup \overline{b} \cap r$

$= \bigcup_n (P_n)$

EXERCISES (18)

1. Derive and simplify the following preconditions:

a. $(y := 0;\ \underline{do}\ y+1 < x \rightarrow y := y+1\ \underline{od})\ \underline{a}\ (y = x - 1)$

b. $x := X;\ y := Y;\ \underline{do}\ y > 0 \rightarrow x := x+1;\ y := y - 1\ \underline{od}\ \underline{a}\ x = X + Y + Z$

c. $y := Y;\ s := 0;\ \underline{do}\ y > 0 \rightarrow y := y - 1;\ s := s + X\ \underline{od}\ \underline{a}\ s = X*Y$

d. $x := X;\ q := 0;\ \underline{do}\ x \geq y \rightarrow x := x - y;\ q := q+1\ \underline{od}\ \underline{a}\ X = q*y+x\ \&\ x < y$

(all variables are nonnegative integers).

We can now solve problems of the form: (19)

given command S and postcondition r,

$? = S\ \underline{a}\ r$

But programmers must solve a different problem:

given postcondition r and precondition p,

$p \Rightarrow ?\ \underline{a}\ r$

e.g. using only +1 and $<$, write a command c which does not change x, and which satisfies:

$0 < x \Rightarrow c\ \underline{a}\ y = x - 1$

This is usually more difficult! We shall need some more theory.

TRIVIALITIES (20)

If p and b are conditions

$p \cap b = p;\ b = p\ \underline{a}\ b = b\ \underline{a}\ p = b;p$

$c;\ (c1 \cup c2) = c;\ c1 \cup c;\ c2$

$(c1 \cup c2);\ c = c1;\ c \cup c2;\ c$

$p \cap (b;\ c\ \underline{a}\ r) = p\ \underline{a}\ (b;\ c\ \underline{a}\ r) = (p;b;c)\ \underline{a}\ r$

$= (p \cap b;\ c)\ \underline{a}\ r = b \cap (p \cap b;\ (c\ \underline{a}\ r))$

if $p \cap b \Rightarrow c1\ \underline{a}\ r$

and $p \cap \overline{b} \Rightarrow c2\ \underline{a}\ r$

then $p \Rightarrow \underline{if}\ b \rightarrow c1,\ c2\ \underline{fi}\ \underline{a}\ r$

THEOREM OF INVARIANCE (22)

Let c be $\underline{do}\ b \rightarrow c1\ \underline{od}$ (for deterministic $c1$)

and let $b \cap p \subseteq c1\,\underline{a}\ p$ ... (1)

(i.e. p is an invariant of c)

then $p \cap (c\ \underline{a}\ \underline{T}) \subseteq c\ \underline{a}\ (\bar{b} \cap p)$

Proof. define $c_0 = \bar{b}$, $c_{n+1} = b;\ c1;\ c_n \cup \bar{b}$

so $c = \bigcup_n c_n$

$$LHS = p \cap (\bigcup_n c_n)\underline{a}\ \underline{T} = p \cap \bigcup_n (c_n\ \underline{a}\ \underline{T}) \qquad \text{distribution}$$

$$= \bigcup_n p \cap (c_n\ \underline{a}\ \underline{T}) \subseteq \bigcup_n (c_n\ \underline{a}\ (\bar{b} \cap p)) \qquad \text{(by lemma)}$$

$$= (\bigcup_n c_n)\ \underline{a}\ (\bar{b} \cap p) = RHS. \qquad \text{distribution}$$

LEMMA (23)

If c is deterministic (1)

and $p \cap b \subseteq c\ \underline{a}\ p$ (2)

and $c_0 = \bar{b}$ and $c_{n+1} = b;c;c_n \cup \bar{b}$ (3)

then $\forall n$ $\underline{p \cap (c_n\ \underline{a}\ \underline{T}) \subseteq c_n\ \underline{a}\ (\bar{b} \cap p)}$ (4)

Proof case-1 $n = 0$: $p \cap (\bar{b}\ \underline{a}\ \underline{T}) = p \cap \bar{b} = \bar{b} \cap (p \cap \bar{b}) = \bar{b}\ \underline{a}(\bar{b} \cap p)$

$$p \cap (c_{n+1} \underline{a}\ \underline{T}) = p \cap (b;c;c_n)\ \underline{a}\ \underline{T} \cup p \cap (\bar{b}\ \underline{a}\ \underline{T}) \qquad \text{by(3)}$$

$$\subseteq b \cap (c\ \underline{a}\ p) \cap (c\ \underline{a}(c_n\ \underline{a}\ \underline{T})) \cup p \cap \bar{b} \qquad \text{by(2)}$$

$$= b \cap (c\ \underline{a}\ (p \cap (c_n\ \underline{a}\ T))) \cup p \cap \bar{b} \qquad \text{by(1)}$$

$$\subseteq b \cap (c\ \underline{a}\ (c_n\ \underline{a}\ (\bar{b} \cap p)))\ \ \bar{b}\ \underline{a}\ (\bar{b} \cap p) \qquad \text{by(4)}$$

$$= (b;c;\ c_n)\ \underline{a}\ (\bar{b} \cap p) \cup \bar{b}\ \underline{a}\ (\bar{b} \cap p)$$

$$= ((b;c;c_n) \cup \bar{b})\ \underline{a}\ (\bar{b} \cap p) \qquad (3)$$

$$= c_{n+1}\ \underline{a}\ (\bar{b} \cap p)$$

VARIANT EXPRESSIONS (24)

Let t be an expression, (always defined)

$$c \ \underline{dec}\ t =_{df} (k := t;\ c)\ \underline{a}\ (0 \leqslant t < k)$$

(where k is a fresh variable).

i.e. the weakest precondition under which the command c <u>decreases</u> value of t.

Theorem 2. Let $c = \underline{do}\ b \rightarrow c1\ \underline{od}$ (deterministic)

$$p \cap b \subseteq c1\ \underline{a}\ p$$

$$p \cap b \subseteq c1\ \underline{dec}\ t$$

then $p \subseteq c\ \underline{a}\ (\overline{b} \cap p)$

Proof define $c_0 = \overline{b}$, $c_{n+1} = b;c;c_n \cup \overline{b}$

by Lemma 2 $p \cap (t \leqslant n) \subseteq c_n\ \underline{a}\ \underline{T}$

$$\therefore\ p = \bigcup_n (p \cap t \leqslant n) \subseteq p \cap \bigcup_n (c_n\ \underline{a}\ \underline{T}) = p \cap c\ \underline{a}\ \underline{T}$$

$$\subseteq c\ \underline{a}\ (\overline{b} \cap p) \qquad \text{by theorem (1)}$$

LEMMA (25)

If $p \cap b \subseteq (k := t;c)\ \underline{a}\ (p \cap t < k)$ for fresh k (1)

and $c_0 = \overline{b}$, $c_{n+1} = b;c;c_n \cup \overline{b}$ (2)

then $\forall n\ \ p \cap t \leqslant n \subseteq c_n\ \underline{a}\ \underline{T}$ (3)

Proof. $p \cap b \subseteq (k := t;c)\ \underline{a}\ t < k \subseteq t > 0$

$\therefore\ p \cap t \leqslant 0 \subseteq \overline{b} = c_0$.

induction step.

$$p \cap t \leqslant n{+}1 = p \cap t = n{+}1 \qquad \vee\ p \cap t \leqslant n$$

$$\subseteq p \cap b \cap t = n{+}1 \cup \overline{b} \qquad \cup \qquad "$$

$$\subseteq b \cap t = n{+}1 \cap (k := t;c)\underline{a}\ (p \cap t < k) \cup \overline{b} \qquad \cup \qquad "$$

$$\subseteq b \cap c\ \underline{a}(p \cap t < n{+}1) \cup \overline{b} \qquad \cup \quad p \cap t \leqslant n$$

$$\subseteq b \cap c\ \underline{a}(c_n\ \underline{a}\ \underline{T}) \cup \overline{b}\ \underline{a}\ \underline{T} \qquad \cup \quad c_n \underline{a}\ \underline{T}$$

$$= c_{n+1}\ \underline{a}\ \underline{T} \cup c_n\ \underline{a}\ \underline{T}$$

$$= c_{n+1}\ \underline{a}\ \underline{T} \qquad \text{since } c_n \subseteq c_{n+1}$$

EXAMPLE 1 (26)

Using only $<$, successor as operators, find c s.t.

$0 < x \Longrightarrow c \ \underline{a} \ y = x - 1$ & c changes only y.

Solution: reformulate postcondition as

$\overline{y + 1 < x} \cap y < x \qquad = \bar{b} \cap p$

and find c1, t s.t.

$y + 1 < x \cap y < x \Longrightarrow c1 \ \underline{dec} \ t$

$c1 \ \underline{a} \ y < x$

try $c1 = y := y + 1$

$t = x - y$

check $y + 1 < x \Longrightarrow (k := x - y;\ y := y + 1) \ \underline{a} \ (x - y < k \cap y < x)$

RHS $= x - y - 1 < x - y$

∴ by theorem 2.

$y < x \Longrightarrow \underline{do} \ y + 1 < x \longrightarrow y := y + 1 \ \underline{od} \ \underline{a} \ y = x - 1$

It remains to find c0 s.t.

$0 < x \Longrightarrow c0 \ \underline{a} \ y < x.$

Using theorem of assignment, c0 is $y := 0$.

EXAMPLE 2 (27)

Using only $<$, successor, and predecessor as operators find c s.t.

$c \ \underline{a} \ x = X + Y$ where c changes only x and y.

Solution: reformulate postcondition as

$\bar{b} \cap p = \overline{0 < y} \cap x + y = X + Y$

check that $x := X;\ y := Y \ \underline{a} \ p$. ... (1)

we need to find c1, t s.t.

$0 < y \cap x + y = X + Y \Longrightarrow (k := t;\ c1) \ \underline{a} \ (t < k \cap p)$

an obvious choice is $t = y$.

$c1 = c2;\ y := y - 1$ ... (2)

where $0 < y \cap p \Rightarrow c2$ a ($y := y - 1$ a $x + y = X + Y$)

i.e. $0 < y \cap (x + y = X + Y) \Rightarrow c2$ a $(x + y - 1 = X + Y)$

obviously $c2 = x := x + 1$ .... (3)

collecting (1), (2), (3), we get:

$x := X$; $y := Y$; do $0 < y \rightarrow x := x + 1$; $y := y - 1$ od

EXAMPLE 3 (28)

Using only $\leq$, +, - , find c s.t.

$Y > 0 \Rightarrow c$ a $q = X \div Y \cap r = X$ modulo Y. c changes only q and r

Reformulate postcondition as

$\overline{Y \leq r} \cap X = q * Y + r$ $(= \bar{b} \cap p)$ (1)

note that ($q := 0$; $r := X$) a p

choose r as variant function.

$b \Rightarrow r := r - Y$ dec r (2)

we need to find c1 changing only q, s.t.

$b \cap p \Rightarrow c1$ a $(X = q * Y + r - Y)$

this is solved by $c1 = q := q + 1$.

$\therefore$ $c = q := 0$; $r := X$; do $Y \leq r \rightarrow q := q + 1$; $r := r - y$ od

Exercise: using only $<$, +, - , find c s.t.

c a $s = X * Y$, and c changes only s and y.