计算的数学理论

（下 册）

ZOHAR MANNA 著

王 冬 生 译

长沙铁道学院科技情报室

一九七九年九月

目　　录

第三章　程序的正确性检验……（1）

引　言……（1）

3—1　流图程序……（1）

3—1·1　部分正确性……（10）

3—1·2　终结性……（22）

3—2　带有阵列的流图程序……（28）

3—2·1　部分正确性……（29）

3—2·2　终结性……（35）

3—3　Algol 类程序……（43）

3—3·1　while 程序……（44）

3—3·2　部分正确性……（46）

3—3·3　全正确性……（53）

文献评论．文献目录．习题．……（62）

第四章　流图格式……（89）

引　言……（89）

4—1　基本概念……（90）

4—1·1　句　法……（90）

4—1·2　字义（说明）……（92）

4—1·3　基本性质……（97）

4—1·4　Herbrand 说明……（108）

4—2　判定问题……（110）

4—2·1　基本性质的不可解性……（113）

4—2·2　自由格式……（117）

4—2·3　树格式……（124）

4—2·4　Ianov 格式……（133）

4－3　用谓词演算形式化……(142)
4－3·1　演　法……(143)
4－3·2　流图程序性质的形式化……(153)
4－3·3　流图格式性质的形式化……(158)

4－4　转换问题……(164)
4－4·1　递归格式……(166)
4－4·2　流图格式与递归格式的比较……(170)
文献评论，文献目录，习题……(180)

第五章　程序的不动点理论……(205)

引　言……(205)

5－1　函数与泛函……(206)
5－1·1　单调函数……(208)
5－1·2　连续泛函……(216)
5－1·3　泛函的不动点……(220)

5－2　递归程序……(225)
5－2·1　计算规则……(226)
5－2·2　不动点计算规则……(235)
5－2·3　递归定义系统……(242)

5－3　检验方法……(245)
5－3·1　分步计算的归纳法……(245)
5－3·2　完全计算归纳法……(254)
5－3·3　不动点归纳法……(258)
5－3·4　结构归纳法……(263)
文献评论，文献目录，习题……(273)

第三章 程序的检验

引言

这一章的目的是叙述检验计算机程序的方法。假定已经给了我们一个程序和关于它的性状的描述。这里所讲的是指描述在程序执行完成时，程序变量间必须满足的关系的特征谓词（后面叫输出谓词）。有时候我们也给出一个输入谓词，它确定为了使程序的执行有意义而必须满足的输入限制。我们的任务是证明程序关于输入和输出谓词是正确的。即对于一切满足输入谓词的程序，我们应该保证它终止并且在程序执行完时输出谓词得到满足。在本章我们将叙述这种关于正确性证明的结构。

为了讨论程序和它们的正确性我们必须规定程序设计语言。我们将考虑没有阵列的（３－１节）和有阵列的（３－２节）流图程序，以及简单的Algol类程序（３－３节）。关于证明递归程序的正确性的方法将在第五章详细讨论。

３－１　流图程序

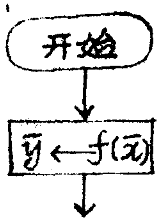
首先，我们考虑一类很简单的流图程序。我们区分三类变量（组成为三个向量）：(1)一个输入向量 $\bar{x}=(x_1, x_2, x_3, \cdots, x_a)$，它由给定的输入变量组成，因而在计算当中永不变动；(2)一个程序向量 $\bar{y}=(y_1, y_2, \cdots, y_b)$，在计算过程中它被用作临时的存储器；(3)一个输出向量 $\bar{z}=\{z_1, z_2, \cdots, z_c\}$，当计算结束时它产生输出值。相应地，我们也区分三类（非空）区域：(1)输入区域 $D_{\bar{x}}$，(2)程序区域 $D_{\bar{y}}$，(3)输出区域 $D_{\bar{z}}$。每一个区域实际上是一些子区域的Cartesian乘积。

$$D_{\bar{x}} = D_{x_1} \times D_{x_2} \times \cdots \times D_{x_a}$$

$$D\bar{y} = D_{y_1}\times D_{y_2}\times\cdots\times D_{y_b}$$
$$D\bar{z} = D_{z_1}\times D_{z_2}\times\cdots\times D_{z_c}$$

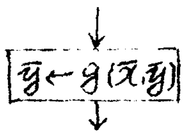
我们也把语句区分成四类，†

1、开始语句



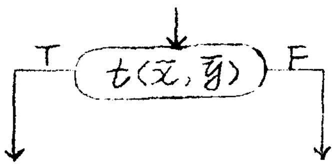
其中 $f(\bar{x})$ 是把 $D\bar{x}$ 变换到 $D\bar{y}$ 的全函数.

2、赋值语句



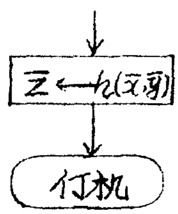
其中 $g(\bar{x},\bar{y})$ 是把 $D\bar{x}\times D\bar{y}$ 变换到 $D\bar{y}$ 的全函数.

3.测试语句



其中 $t(\bar{x},\bar{y})$ 是 $D\bar{x}\times D\bar{y}$ 上的一个全谓词.

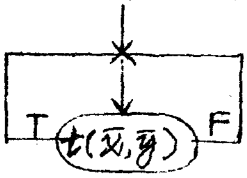
4.仃机语句



其中 $h(\bar{x},\bar{y})$ 是把 $D\bar{x}\times D\bar{y}$ 变换到 $D\bar{z}$ 的全函数. 一个流图程序仅仅是由这些语句（只有一个开始语句）构成的任意,

† 又及，我们显然允许使用把两个以上的箭头组合成一个箭头的复合语句.

“流图”，使得每个赋值语句或测试语句在由开始语句到某个仃机语句的路径上．换句话说，不允许流图程序包含如下的“静止端点”的测试语句．



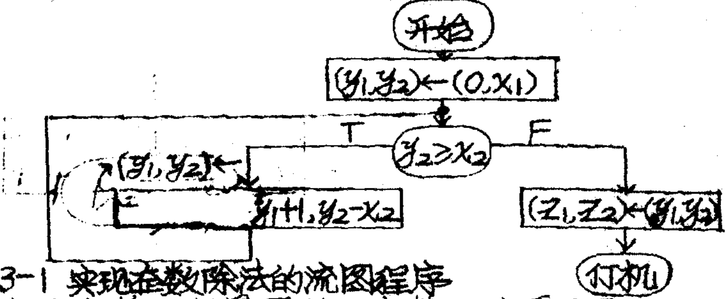


图3-1 实现整数除法的流图程序

给定一个流图程序P和输入向量$\bar{x}$的一个输入值$\bar{\xi}\in D_{\bar{x}}$，便可执行程序．即从开始语句起执行程序，把$f(\bar{\xi})$赋给$\bar{y}$作为初值，然后顺着箭头逐句往下执行．每当执行到赋值语句时，$\bar{y}$的值换成了$g(\bar{x},\bar{y})$的值，其中$g(\bar{x},\bar{y})$中的$\bar{x}$和$\bar{y}$是代换之前的最新值．每当执行到测试语句时，根据$t(\bar{x},\bar{y})$的当前值是真还是假来选择程序的执行顺序．当$t(\bar{x},\bar{y})$的当前值为真，则程序沿T分支往下执行；若其为假，则沿着F分支往下执行．通过测试语句，$\bar{y}$的值不会改变．当程序执行到仃机语句，便把$h(\bar{x},\bar{y})$的当前值$\bar{\zeta}$赋给$\bar{z}$，并说$P(\bar{\xi})$被定义且记作$P(\bar{\xi})=\bar{\zeta}$．反之若永远达不到仃机语句，这时我们说$P(\bar{\xi})$是无定义的．换言之，程序P应看作一个把$D_{\bar{x}}$变换到$D_{\bar{z}}$的部分函数$\bar{z}=P(\bar{x})$．

例如，设$x_1\geq 0$，$x_2>0$且都是整数．$x_1\div x_2$的商为z_1，余数为z_2．即$z_1=div(x_1,x_2)$，$z_2=rem(x_1,x_2)$．实现整数除法的流图程序画在图3－1上．这里，$\bar{x}=\{x_1,x_2\}$，$\bar{y}=\{y_1,y_2\}$，$\bar{z}=\{z_1,z_2\}$而$D_{\bar{x}}=D_{\bar{y}}=D_{\bar{z}}=\{$全体整数偶$\}$．

注意，$(y_1,y_2)\leftarrow(0,x_1)$的意思是以0代$y_1$，以$x_1$代$y_2$，同样地，$(y_1,y_2)\leftarrow(y_1+1,y_2-x_2)$意味着以$y_1+1$代$y_1$，以$y_2-x_2$代$y_2$．一般地，我们将用记号$(y_1,y_2,\cdots,y_n)\leftarrow(g_1(\bar{x},\bar{y}),g_2(\bar{x},\bar{y}),\cdots,g_n(\bar{x},\bar{y}))$表示同时以$g_i(\bar{x},\bar{y})$代$y_i$（$1\leq i\leq n$），也就是说，在任何$y_i$改变

之前，要把所有的 y_i 的值先示好。例如，若 $y_1=1$ 而 $y_2=2$，经过赋值 $(y_1, y_2) \longleftarrow (y_1+1, y_1+y_2)$ 之后，$y_2=3$ 而不是 $y_2=4$。赋值语句 $(y_1, y_2) \longleftarrow (y_2, y_1)$ 起到交换 y_1 和 y_2 的内容的作用。

流图程序的检验依赖于两个给定的谓词。

1、输入谓词 $\varphi(\bar{x})$．它是 $D_{\bar{x}}$ 上的全谓词，并且描述 $D_{\bar{x}}$ 上那些可以用作输入的元素．换句话说，在程序的执行当中，我们仅对 $D_{\bar{x}}$ 中满足谓词 $\varphi(\bar{x})$ 的元素有兴趣．在特殊情况下，我们可能对 D_x 中的一切元素都有兴趣，这时就令 $\varphi(\bar{x})$ 为T，即 $\varphi(\bar{x})$ 对 $D_{\bar{x}}$ 的一切元素为真。

2、输出谓词 $\psi(\bar{x}, \bar{z})$ 是定义在 $D_{\bar{x}} \times D_{\bar{z}}$ 上的全谓词，它描述在程序执行完毕时，输入变另和输出变另之间必须满足的关系。

下面我们给出几个定义

1、如果对于每个使 $\varphi(\bar{\xi})$ 为真的输入另 $\bar{\xi}$，程序的计标都终结，我们就说P在 φ 上终结。

2、若对每一个使得 $\varphi(\bar{\xi})$ 为真并且程序的计标终结的 $\bar{\xi}$，$\psi(\bar{\xi}, P(\bar{\xi}))$ 是真的，那么我们就说P关于 φ 和 ψ 是部分正确的。

3、P关于 φ 和 ψ 是全正确的，如果对于每一个使得 $\varphi(\bar{\xi})$ 为真的 $\bar{\xi}$，程序的计标终结并且 $\psi(\bar{\xi}, P(\bar{\xi}))$ 为真．

于是，在部分正确性中我们"不关心"终结的问题，但是在全正确性中终结问题是本质的。对于一个给定的输入谓词 $\varphi(\bar{x})$ 和一个输出谓词 $\psi(\bar{x}, \bar{z})$ 去检验程序是旨在证明它关于 φ 和 ψ 是全正确的，然而，通常采用两个独立的步骤来证明是十分方便的，首先证明关于 φ 和 ψ 的部分正确性，然后证明在 φ 上程序终结。

现在我们介绍关于证明流图程序即是部分正确的又是终结的方法。让我们来证明前面介绍的除法程序的正确性，首先证明这个程序关于输入谓词

$$\varphi(x_1, x_2):\quad x_1 \geq 0 \wedge x_2 \geq 0$$

和输出谓词

$$\psi(x_1, x_2, z_1, z_2):\quad x_1 = z_1 x_2 + z_2 \wedge 0 \leq z_2 < x_2$$

是部分正确的。然后证明这个程序在

$$\varphi'(x_1, x_2):\quad x_1 \geq 0 \wedge x_2 > 0$$

上是终结的。读者注意，谓词φ允许我们考虑x_1和x_2都是非负数的情形，而谓词ψ实际上是在数除法的定义。因为该程序关于φ和ψ是部分正确的并且在φ'上终结，这就推出该程序关于φ'和ψ是全正确的。注意φ和φ'的差别；在讨论终结时，为除去$x_2=0$的情况，它是不可少的，因为对于$x_2=0$该程序不终结。

部分正确性　为了证明除法程序关于φ和ψ的部分正确性（见图3—2），我们在点A附上一个输入谓词φ在点C附上一个输出谓词ψ。程序检验中的主要问题是怎样处理循环。在B点把程序截断，那么这个程序的循环便变得好处理了。经过截断后，程序的流被分解成三条通路：第一个通路是从A到B（箭头1和2）；第二条通路是从B出发经过循环回到B（箭头3，4和5）；第三条通路是B到C（箭头3，6，7）。我们把这三个通路分别区分为α（开始），β（循环）和γ（停机）。程序从开始语句到停机语句执行的过程是：沿着通路α，经过绕β的若干次（可能0次）循环，最后经过γ达到停机语句。于是，一切执行过程都被这三条通路"复盖"。

为了证明程序的部分正确性，首先必须找出在截断点B处描述程序变量之间的关系的一个谓词$P(x_1, x_2, y_1, y_2)$。符合这个目的的恰当的谓词可取作：

$$x_1 = y_1 x_2 + y_2 \wedge y_2 \geq 0$$

一旦得到这个谓词，就已经用三条通路把程序复盖了（每一条通路从一个谓词开始和一个谓词结束）。通过检验三条通路α，β

和γ中的每一条就能证明程序的部分正确性．具体怎么检验一个通路？那就要证明如果它的初始谓词对$\bar{x}$和$\bar{y}$的某个值为真并且执行这个通路的程序，那末最后的谓词对于$\bar{x}$和$\bar{y}$的新值也为真．

在整数除法的程序中，α通路的检验确立$P(x_1, x_2, y_1, y_2)$在第一次进入程序的循环部分时为真（假定满足程序的输入谓词）．通路β的检验表明，如果$P(x_1, x_2, y_1, y_2)$第一次在循环的入口为真，那么它第二次亦为真；如果它第二次为真，那么它第三次亦为真；等等．通路γ的检验表明，当以$P(x_1, x_2, y_1, y_2)$为真而离开循环时，那么程序的输出谓词为真．换句话说，通过对α和β通路的检验保证$P(x_1, x_2, y_1, y_2)$有这样的性质．每当程序的计标达到截断点B，$P(x_1, x_2, y_1, y_2)$对x_1，x_2，y_1和y_2的当前值是真的．因此，通路γ的检验蕴涵着每当程序的计标达到点C，程序的输出谓词为真．

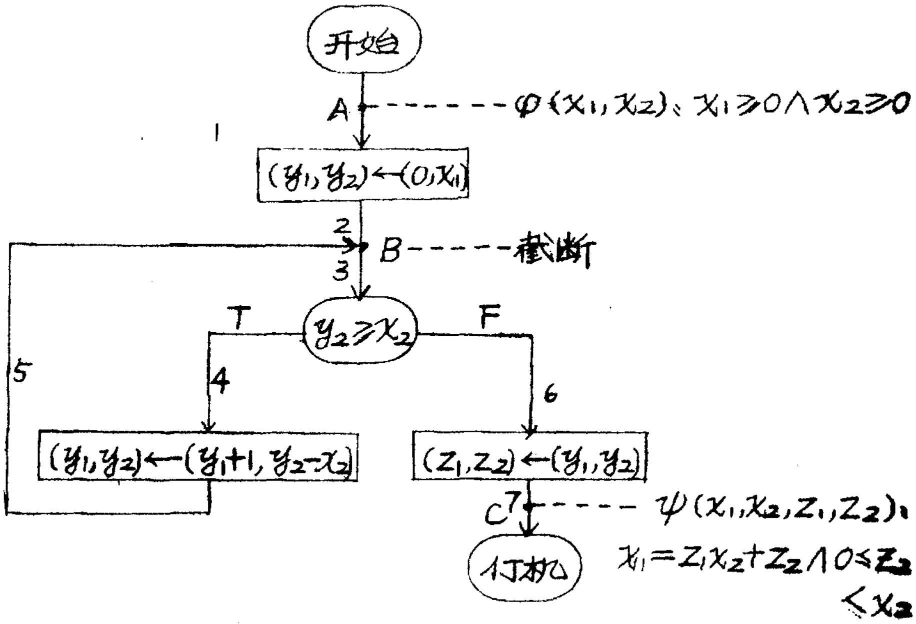
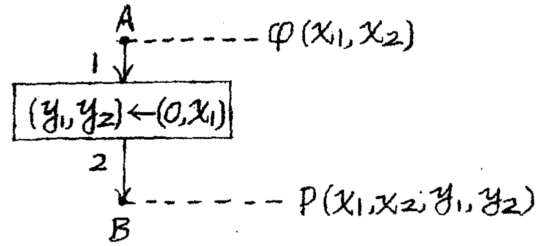


图3−2　具有输入输出谓词与
实现整数除法的流图程序

为了完成除法程序部分正确性的证明，我们必须检验通路α，β和γ．一条通路的检验分两步进行．首先，我们依照给定的谓

词对于每个通路构造一个检验条件，然后我证明这个条件．构造一条通路的检验条件．通常采用经过通路向后移动，依次考虑每一个语句的方法来实现的．通路α．考虑图3－2的第一条通路

通路α．考虑图3－2的第一条通路α．



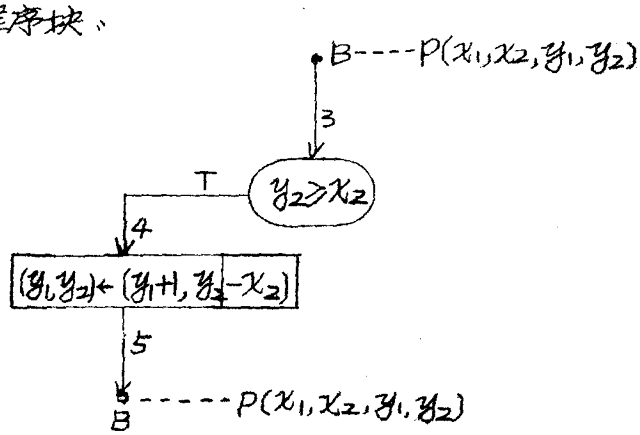
在这个通路中 $\varphi(x_1,x_2)$ 是初始谓词，$P(x_1,x_2,y_1,y_2)$ 是最后谓词．根据前面讲过的检验一条通路的原则，那么这儿的检验条件应该是若 φ 为真那么 P 也为真．但是 P 为真是什么意思？是语句 $(y_1,y_2)\longleftarrow(0,x_1)$ 执行之前 $P(x_1,x_2,y_1,y_2)$ 为真呢还是该语句执行之后为真？答案是明显的，即

$$\varphi(x_1,x_2)\supset P(x_1,x_2,0,x_1)$$

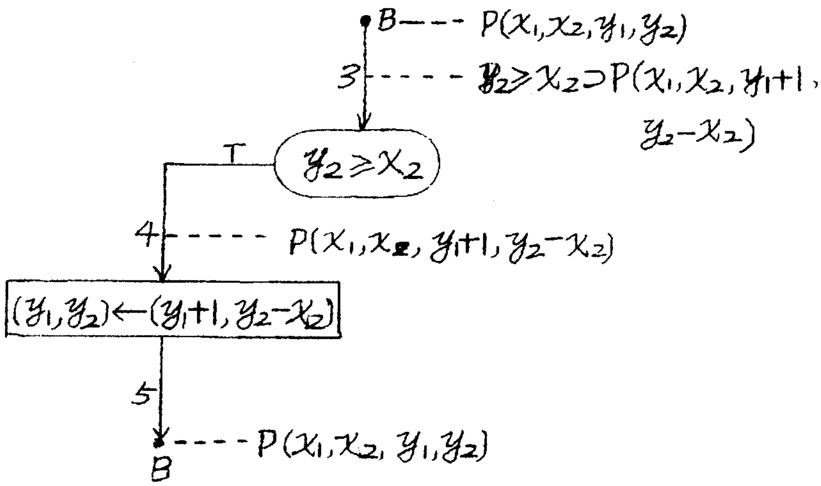
P 中 y_1 的一切出现已用 0 代替，y_2 的一切出现已用 x_1 代替．上面的检验条件是：

$$[x_1\geq 0\wedge x_2\geq 0]\supset[x_1=0\cdot x_2+x_1\wedge x_1\geq 0]$$

通路β．为了构造通路β的检验条件，必须处理测试语句．当我们沿着通路β前进（测试语句发现 $y_2\geq x_2$），这个通路可以表示为加注解的流图程序块．



通过测试语句，再往回走，这时导出谓词 $P(x_1,x_2,y_1,y_2)$ 在箭头4处产生 $P(x_1,x_2,y_1+1,y_2-x_2)$。虽然测试语句不改变程序变量的任何值，但是它提供了一个有用的附加信息。因为通过测试之后，显然有 $y_2\geq x_2$. 为了处理这种情况，通常对赋值语句提出同样的问题。在箭头3处什么谓词必须为真才能使得当控制取道测试语句的T分支时（即当 $y_2\geq x_2$ 为真时）谓词 $P(x_1,x_2,y_1+1,y_2-x_2)$ 在箭头4处为真？在这种情况下，回答是简单的：$y_2\geq x_2 \supset P(x_1,x_2,y_1+1,y_2-x_2)$. 于是 β 通路的完全的分析是：



在这种情况下要证明的检验条件是

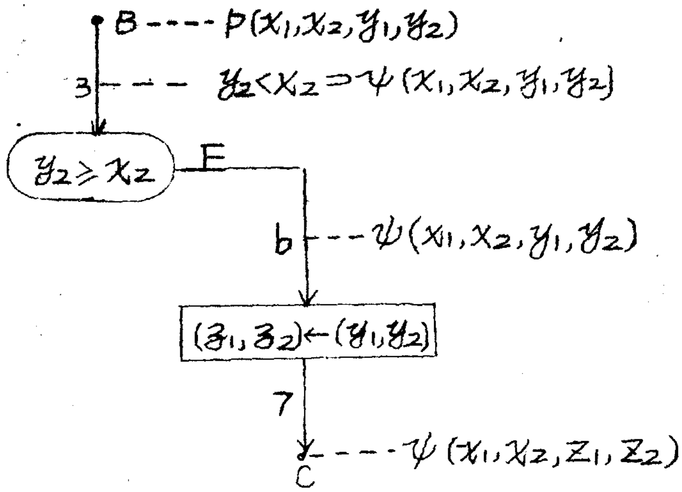
$$P(x_1,x_2,y_1,y_2)\supset[y_2\geq x_2\supset P(x_1,x_2,y_1+1,y_2-x_2)]$$

或等价地，

$$P(x_1,x_2,y_1,y_2)\wedge y_2\geq x_2\supset P(x_1,x_2,y_1+1,y_2-x_2)^{\dagger}$$

通路 γ。对 γ 通路的分析结果，画在下图中。

✦在本章中，一律把 $A_1\wedge A_2\supset B$ 理解为 $(A_1\wedge A_2)\supset B$，一般地，把 $A_1\wedge A_2\wedge A_3\wedge\cdots\wedge A_n\supset B$ 理介为 $(A_1\wedge A_2\wedge\cdots\wedge A_n)\supset B$.



所要证明的检验条件是：

$$P(x_1,x_2,y_1,y_2) \supset [y_2<x_1 \supset \psi(x_1,x_2,y_1,y_2)]$$

或等价地，

$$P(x_1,x_2,y_1,y_2)\wedge y_2<x_2 \supset \psi(x_1,x_2,y_1,y_2)$$

因此，我们已经形成了三个检验条件：

$$\varphi(x_1,x_2) \supset P(x_1,x_2,0,x_1) \qquad (\alpha)$$

$$P(x_1,x_2,y_1,y_2)\wedge y_2\geq x_2 \supset P(x_1,x_2,y_1+1,y_2-x_2) \qquad (\beta)$$

$$P(x_1,x_2,y_1,y_2)\wedge y_2<x_2 \supset \psi(x_1,x_2,y_1,y_2) \qquad (\gamma)$$

其中，

$\varphi(x_1,x_2)$ 是 $x_1\geq 0 \wedge x_2 \geq 0$

$P(x_1,x_2,y_1,y_2)$ 是 $x_1=y_1x_2+y_2 \wedge y_2\geq 0$

$\psi(x_1,x_2,y_1,y_2)$ 是 $x_1=z_1x_2+z_2 \wedge 0\leq z_2<x_2$

读者自己可验证，对于所有的整数x_1，x_2，y_1和y_2，这三个检验条件的真值是true；因此，这个程序关于φ和ψ是部分正确的。

终结　到目前为止，我们仅部分地检验了我们的程序：我们已经证明了一旦计算程序终结（即达到一个停机语句），那么输出

谓词取真值 true ）然而，我们尚未最后证明计标程序确实终结，因此，为了完成除法程序的检验，我们还须证明程序的终结性。

现在我们来证明对于每一输入 x_1 和 x_2，该程序终结，其中

$$\varphi'(x_1,x_2):\quad x_1\geqslant 0 \wedge x_2>0$$

（再注意，对于 $x_2=0$ 该程序不终结。）首先我们证明谓词

$$q(x_1,x_2,y_1,y_2):\quad y_2\geqslant 0 \wedge x_2>0$$

具有这样的性质。在计标的过程当中，每当我们达到 B 点，$q(x_1,x_2,y_1,y_2)$ 关于这些变量的当前值取真值 true，为此目的，我们必须证明下列两个检验条件：

$$\varphi'(x_1,x_2)\supset q(x_1,x_2,0,x_1) \qquad (\alpha)$$

$$q(x_1,x_2,y_1,y_2)\wedge y_2\geqslant x_2\supset q(x_1,x_2,y_1+1,y_2-x_2) \quad (\beta)$$

也就是，

$$(x_1\geqslant 0\wedge x_2>0)\supset(x_1\geqslant 0\wedge x_2>0)$$

$$(y_1\geqslant 0\wedge x_2>0)\wedge y_2\geqslant x_2)\supset(y_2-x_2\geqslant 0\wedge x_2>0)$$

然而，这两式显然取真值 true。

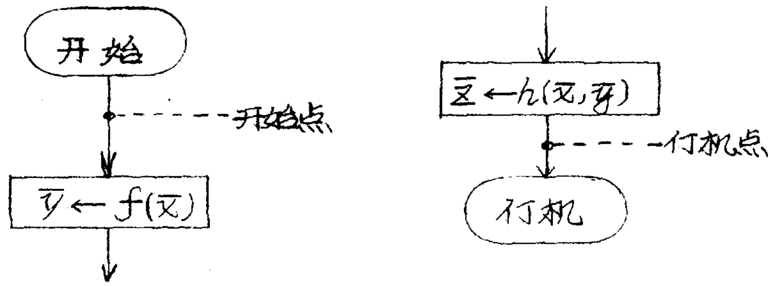
现在我们观定到：因为在 B 点我们总有 $x_2>0$，所以每当我们从 B 点出发经 β 通路回到 B 点绕圈时，每转一圈 y_2 的值减小一次。同时我们知道在 B 点 $y_2\geqslant 0$。由于不存在无限的自然数严格递减序列，所以我们不会无限地循环；换句话说，这种计标必定终结。利用这种办法，我们在 3—1·2 节中将讲述一个关于检验程序终结性的一般技术。

3—1·1 部分正确性

让我们推广在检验除法程序时所阐明的技术。假设，给定了一个流图程序 P，一个输入谓词 φ 和一个输出谓词 ψ；我们将按如下步骤去证明 P 关于 φ 和 ψ 是部分正确的。

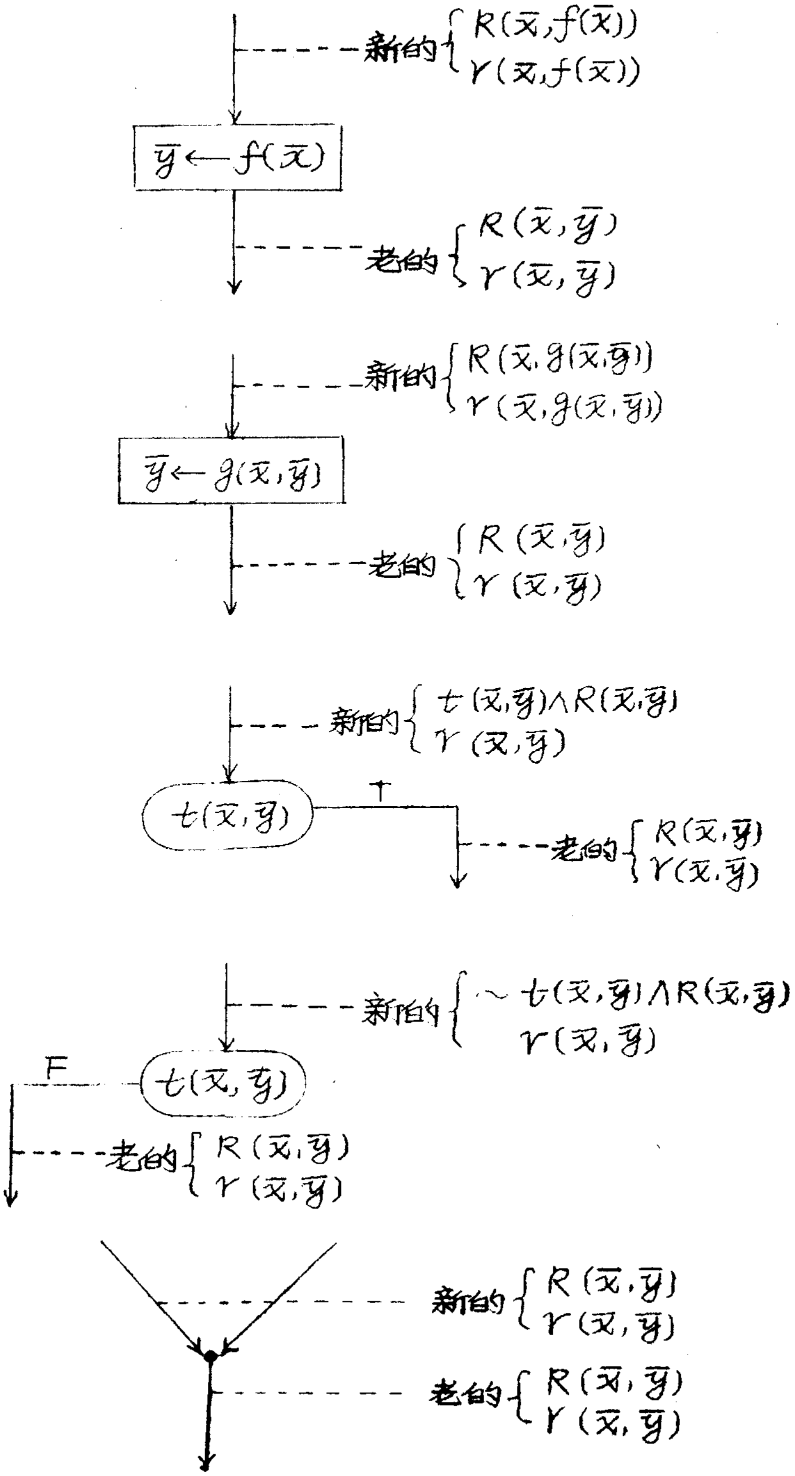
第一步　隔离　在这一步中，我们通过在流图的箭头处选取一个有限点集（其中的点叫做截断点）把程序的各个循环部分截断，

使得每一个循环部分至少包含一个这样的截断点，从而达到了隔离的目的。另外再附加两个点到这个截断点集上，一个是所谓开始点，它位于从开始框引出的箭头上；另一个是所谓停机点，它位于引向停机框的箭头上。



我们只考虑以截断点开头和结尾而其中间再没有截断点的通路。因为限定了每个循环包含一个截断点，所以每条这样的通路是有限的，而且只能存在有限多个通路。设α是由截断点i引向j的通路（注意，正如除法程序的例子那样，$i=j$是可能的）。在下面的讨论中，我们将使用$R_\alpha(\bar{x},\bar{y})$和$r_\alpha(\bar{x},\bar{y})$这两个谓词。前者表示这个将被通过的通路的条件，而后者描述：当该通路被通过时招致$\bar{y}$的值的某种变换。于是$R_\alpha(\bar{x},\bar{y})$是$D_{\bar{x}}\times D_{\bar{y}}$上的谓词，而$r_\alpha(\bar{x},\bar{y})$是把$D_{\bar{x}}\times D_{\bar{y}}$变换到$D_{\bar{y}}$的函数。无论$R_\alpha$或$r_\alpha$通常都是通过α中的函数和测试来表达的；推导它们的一个简单的办法是使用反向代换技术（backward-substitution technique）。

起初，$R(\bar{x},\bar{y})$是对T建立的而$r(\bar{x},\bar{y})$是对$\bar{y}$建立的并且两者都附属于截断点j；然后向着截断点i向回移动，逐步用R和r的老值构造新的R和r，最后在截点i得到的R和r即为所求的R_α和r_α。关于在每一步中构造新的R和r的法则是：



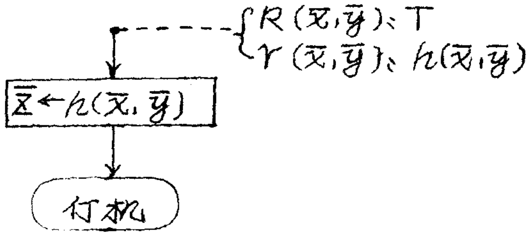
新的 $\begin{cases} R(\bar{x}, f(\bar{x})) \\ r(\bar{x}, f(\bar{x})) \end{cases}$
$\bar{y} \leftarrow f(\bar{x})$
老的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
新的 $\begin{cases} R(\bar{x}, g(\bar{x}, \bar{y})) \\ r(\bar{x}, g(\bar{x}, \bar{y})) \end{cases}$
$\bar{y} \leftarrow g(\bar{x}, \bar{y})$
老的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
新的 $\begin{cases} t(\bar{x}, \bar{y}) \wedge R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
$t(\bar{x}, \bar{y})$
T
老的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
新的 $\begin{cases} \sim t(\bar{x}, \bar{y}) \wedge R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
F
$t(\bar{x}, \bar{y})$
老的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
新的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$
老的 $\begin{cases} R(\bar{x}, \bar{y}) \\ r(\bar{x}, \bar{y}) \end{cases}$

作为例子，我们考虑图3－3所描述的α通路的反向代换技术，我们从截断点j出发，R的开始值为T，r的开始值为$\bar{y}$，逐步进行反向代换，最后在点i处得到．

$R_\alpha(\bar{x},\bar{y})$：$t_1(\bar{x},g_1(\bar{x},\bar{y}))\wedge\sim t_2(\bar{x},g_2(\bar{x},g_1(\bar{x},\bar{y})))$

$r_\alpha(\bar{x},\bar{y})$：$g_3(\bar{x},g_2(\bar{x},g_1(\bar{x},\bar{y})))$

在j点为停机点的特殊情况下，r被赋给初值$\bar{z}$，R被赋给初值T；然后向后移动的第一步中我们得到



然后连续进行向后移动的过程，正如先前所描述的那样．在i是开始点这一特殊情况下，无论是$R_\alpha(\bar{x},\bar{y})$还是$r_\alpha(\bar{x},\bar{y})$都不包含$\bar{y}$的任何出现；$R_\alpha$是$D_{\bar{x}}$上的谓词而$r_\alpha$是把$D_{\bar{x}}$变换到$D_{\bar{y}}$的函数．

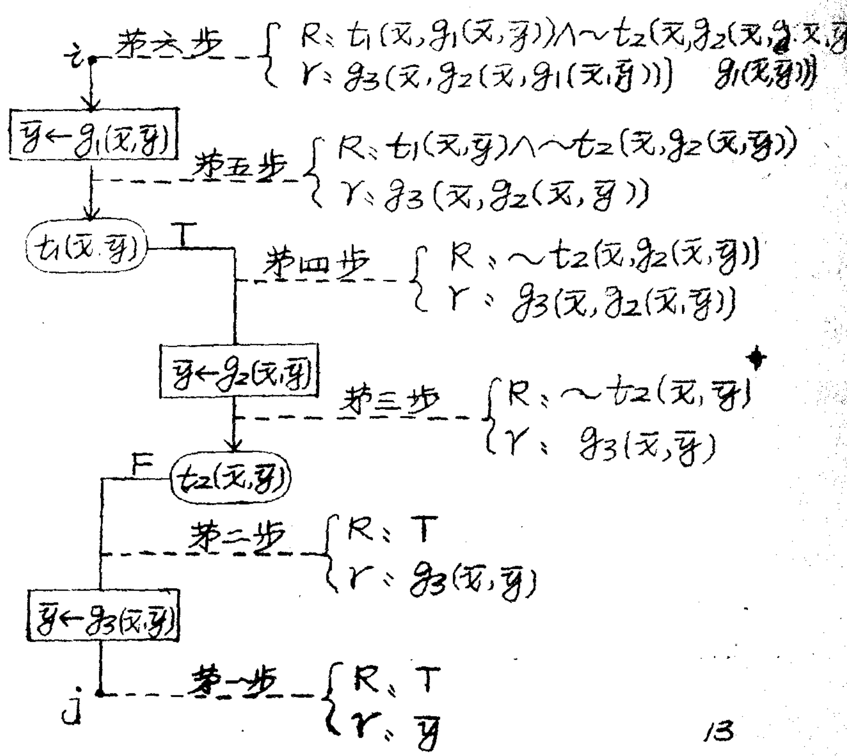


图3－3 反向代换技术．

† 注意，我们已经利用了 $\sim t_2(\bar{x},\bar{y}) \wedge T$ 逻辑地等价于 $\sim t_2(\bar{x},\bar{y})$ 这一事实。

第二步　归纳断言　　第二步是把程序的每一截断点 i 和一个谓词 $P_i(\bar{x},\bar{y})$（通常叫做归纳断言）联系起来，其目的是使得在这点 i 的各变量之间的关系特征化；即，$P_i(\bar{x},\bar{y})$ 将具有这样的性质，每当程序执行到点 i，那时 $P_i(\bar{x},\bar{y})$ 必须关于 $\bar{x}$ 和 $\bar{y}$ 在这一点的当前值取真值 true。输入谓词 $\varphi(\bar{x})$ 附系于开始点，而输出谓词 $\psi(\bar{x},\bar{z})$ 附系于停机点。

第三步　检验条件　　第三步是对于每一个从 i 引向 j 的通路 α 构造检验条件。

$$\forall\bar{x}\forall\bar{y}[P_i(\bar{x},\bar{y}) \wedge R_\alpha(\bar{x},\bar{y}) \supset P_j(\bar{x},r_\alpha(\bar{x},\bar{y}))]$$

这个条件简单地阐明：如果 P_i 关于 $\bar{x}$，$\bar{y}$ 的某些值是真的，而 $\bar{x}$ 和 $\bar{y}$ 是这样的，以致于根据它们在点 i 的值，通路 α 将确实被选择，那么 P_j 关于 $\bar{x}$，$\bar{y}$ 在穿过通路 α 后的值是 true。

在 j 点是停机点的特殊情况下，检验条件是：

$$\forall\bar{x}\forall\bar{y}[P_i(\bar{x},\bar{y}) \wedge R_\alpha(\bar{x},\bar{y}) \supset \psi(\bar{x},r_\alpha(\bar{x},\bar{y}))]$$

而在 i 是开始点的情况下，检验条件是：

$$\forall\bar{x}[\varphi(\bar{x}) \wedge R_\alpha(\bar{x}) \supset P_j(\bar{x},r_\alpha(\bar{x}))]$$

最后一步是去证明所有这些检验条件对于我们的归纳断言的选择在逻辑上是真的。因为证明这些检验条件蕴涵着每一个附系于截断点上的谓词具有这样的性质，每当程序执行到这一点，该谓词关于各变量的当前值取真值 true；特别地，当程序执行到停机点，$\psi(\bar{x},\bar{z})$ 关于 $\bar{x}$ 和 $\bar{z}$ 的当前值取真值 true。换句话说，检验条件的证明蕴涵着给定的程序 P 关于 φ 和 ψ 是部分正确的。

作为总结，我们有下面的定理。