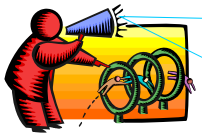
Programming Methodology
程序设计方法学

第2章 程序的控制结构

主讲：刘悦
yliu@staff.shu.edu.cn

回顾



学科地位
程序设计语言和程序设计方法
程序设计方法产生与发展
程序设计方法学的定义与意义
结构化程序设计及其讨论的主要问题

本章目标

？有哪些控制结构
?如何得到良结构的程序

主要内容

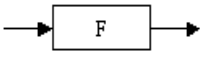
- 什么是结构化程序
- 结构化定理
- 一些新的控制结构

内容线索

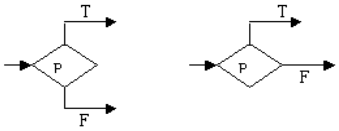
- 什么是结构化程序
  - 流程图程序
  - 正规程序
  - 基本程序
  - 结构化程序
- 结构化定理
- 一些新的控制结构

流程图程序

- **流程图**是一个描述程序的控制流程和指令执行情况的有向图。
  - 1）**函数结点**：有一个入口和一个出口线的结点；



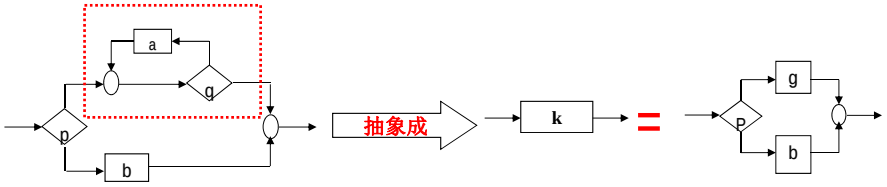
  - 2）**谓词结点**：有一个入口和两个出口线，且它不改变程序的数据项的值的结点；（只作判断，不做计算）

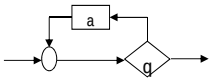


  - 3）**汇点**：两个入口和一个出口线，且它不执行任何运算的结点。

正规程序

- 抽象成：函数结点，概括了该正规程序对数据进行的运算和测试的总的作用。
- 组成：正规子程序。



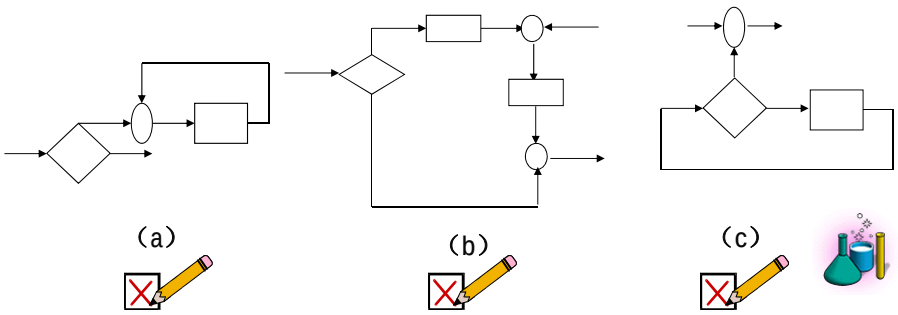


g

g,b是k的正规子程序，函数结点a又是g的正规子程序

同步练习

- 判断一个程序是否为正规程序的标准
  - 首先是流程图程序，且满足：
    - 1）具有一个入口线和一个出口线；
    - 2）对每一个结点，都有一条入口线到出口线的通路通过该结点。



基本程序

- 一个正规程序，如果不包含多于一个结点的正规子程序，称为基本程序。 ?
  - 一个不可再分解的正规程序
  - 仅含一个正规子程序

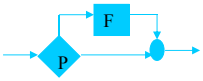
七种基本程序

1、顺序结构

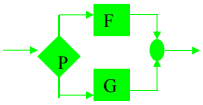
函数

序列 ?

2、选择结构

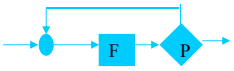


if-then

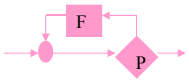


if-then-else

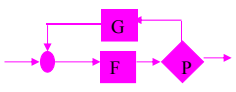
3、循环结构



do-until



while-do



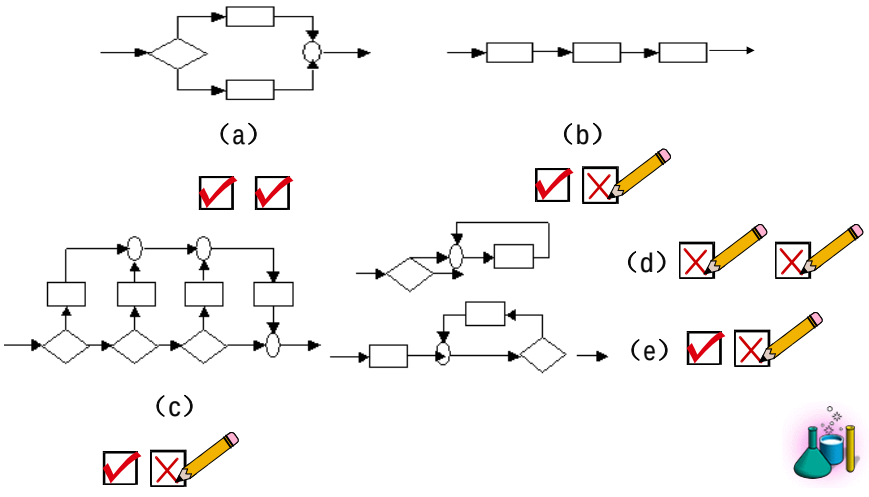
do-while-do

基集合

- 为了构造一个程序，可以只使用七种基本程序中的一部分。
  - 基集合：用以构造程序的基本程序的集合。
  - 例如：
    - {序列，if-then-else，while-do}
    - 或
    - {序列，if-then-else，do-until}
    - …

同步练习

- 判断下列程序是正规程序，还是基本程序？



结构化程序

- 如果一个基本程序的函数结点用另一个基本程序替换，就产生了一个新的正规程序，这样的程序称为复合程序。
- 一个复合程序的规模可大可小。它的复杂程度依赖于所使用的基集合。
  - 例如：
    - 基集合{序列，if-then-else }产生一个无循环的程序类
    - 基集合{序列，do-until }产生的程序类是由基集合{序列， if-then ， while-do}产生的程序类的子集

定义 抽象地说，由基本程序的一个固定的基集合构造出的复合程序称为结构化程序。

结构化程序就是我们通常说得好结构的程序。

主要内容

- 什么是结构化程序
- 结构化定理
  - 程序函数
  - 结构化定理
  - 递归结构程序
- 一些新的控制结构

程序函数

- 程序函数
  - 已知一正则程序P，对于每一个初始的数据状态X，若程序是终止的，那么有确定的最终数据状态Y。如果对于每一个给定的X，值Y是唯一的，那么所有的有序对的集合{（X,Y）}定义了一个函数。我们称这个函数为程序P的程序函数，记为[P]。
    - 例如： 程序P: t:=x; x:=y; y:=t; 程序函数为：
      - {（(x,y,t),(y,x,x)）}
        - 对于任意给定的X: (x,y,t) ,程序执行结果为 Y: (y,x,x)

程序函数的表示形式

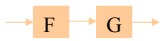
- 程序函数表示形式
  - 有序对、数据赋值以及条件规则等形式
    - 例如：[if x ≤ y then z:=x else z:=y fi]
    - ={((x,y,z),(x,y,min(x,y)))} / /有序对的形式
    - {(x,y,z)|x ≤y→z:=x ∧ x>y→z:=y} //条件规则的形式
    - (z:=min(x,y))//数据赋值的形式

基本程序所对应的程序函数…

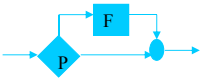
- 函数： [f]={(x,y)|y=f(x)}

g∘f(x)表示函数的复合，即g(f(x))

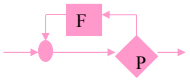
- 序列： [f; g]={(x,y)|y=g∘f(x)}



- IF-THEN: [if-then]={(x,y)|p(x) →y=f(x)|¬p(x) →y=x}



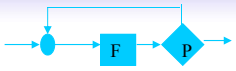
- WHILE-DO:
  - [while-do]={(x,y)| ∃ k ≥ 0((∀j,1≤j<k)(p∘f^j(x)) ∧ ¬p∘f^k(x)→y= f^k(x)}



f^i(x)表示同一函数的多重复合，即 f^0(x)=x， f^k(x)=f∘f^{k-1}(x), k=1,2,…

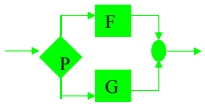
…基本程序所对应的程序函数

- DO-UNTIL:



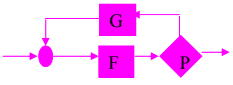
  - [do-until]={(x,y)| ∃ k > 0((∀ j,1 ≤ j<k)(¬p∘f^j(x)) ∧p∘f^k(x)→y= f^k(x))

- IF-THEN-ELSE :



  - [if-then-else]={(x,y)|p(x) →y=f(x)|¬p(x) →y=g(x)}

- DO-WHILE-DO:



  - [do-while-do]={(x,y)| ∃ k ≥ 0((∀ j, 1 ≤ j<k)(p∘f(g∘f)^j(x)) ∧ ¬p∘f(g∘f)^k(x)→y= f∘(g∘f)^k(x))

程序函数是对程序功能的一个精确的描述。

定义 如果程序P_1和P_2有相同的程序函数，称它们是函数等价的，或者简称是等价的。

同步练习

- 写出下列程序的程序函数
  - 1、[while x>0 do x:=x-1 od]
    - ={(x,min(0,x))}=(x:=min(0,x))
  - 2、[x:=x+y;y:=x-y]
    - ={((x,y),(x+y,x))}=(x, y:=x+y,x)
  - 3、[do x:=x+1 until x>y od]
    - ={((x,y),(max(x+1,y+1),y))}=(x:=max(x+1,y+1))

非结构程序

- M.H.Williams把非结构程序归纳为五种基本的成分

(1)异常的选择通路；
(2)循环多出口点；
(3)循环多入口点；
(4)重叠的循环；
(5)并行的循环。

- 可以证明，如果一个流程图是非结构化的，则它至少包含这五种非结构成分中的一种。
  - 因为包含这五种结构的任何一种流程图，都不能简化成单一的处理单元

结构化定理…

- 2） 结构化定理
  - 证明任何一个正规程序都可以用一个等价的结构化程序表示出来
  - 结构化定理
    - 任一正规程序都可以函数等价于一个由基集合{序列，IF-THEN-ELSE，WHILE-DO }产生的结构化程序。
      - 证明过程提供了一种将正规程序转化为结构化程序的方法

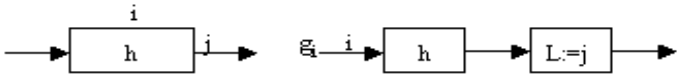
Next Page...

非结构化———〉结构化

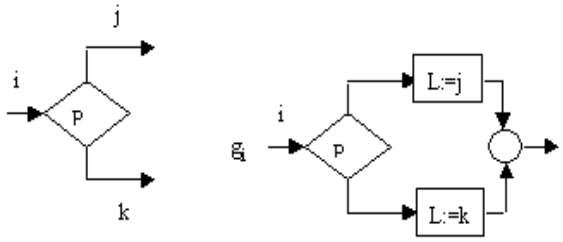
考察任何一个正规程序。

首先，从程序的入口处开始给程序的函数结点和谓词结点编号，编号为1，2，……，n(如果是汇点，那么沿汇点的出口线继续考察，直到找到第一个函数结点或谓词结点)。同时，将每一个函数及谓词结点的出口线用它后面的节点的号码进行编号。如果他后面没有函数或谓词结点，即该结点的出口线直接或通过汇点和程序的出口相连时，出口线编号为0

其次，对原程序中的每一个编号为i, 出口线编号为j的函数结点h，构造一个新的序列程序g_i。

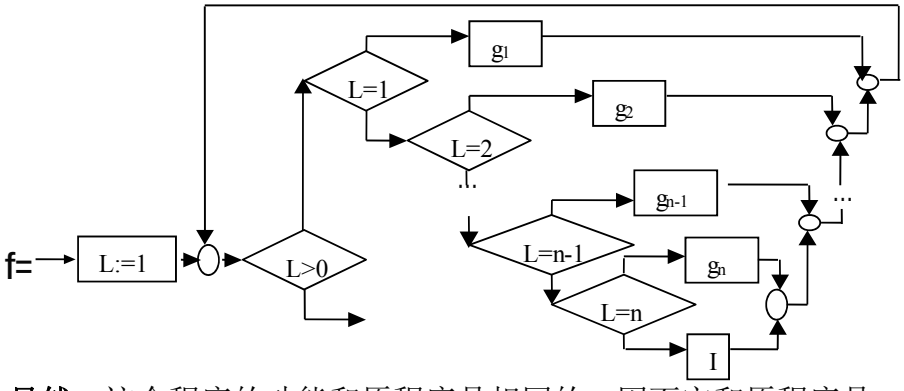


其中，L作为计数器。类似地，对每一点编号为i, 出口线编号为j,k的谓词结点，构造一个新的if-then-else程序g_i.



2 - 23

最后，利用已经得到的一些g_i（i=1,2,……,n）按下图的形式构造一个while-do循环，这个循环的循环体是一个对L从1到n进行测试的嵌套的if-then-else程序（最内层的I表示恒等函数）。
如果I_x={<x,x>|x∈X},则称函数I_x :X → X为恒等函数。

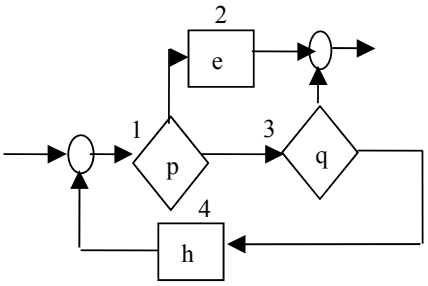


显然，这个程序的功能和原程序是相同的，因而它和原程序是函数等价的。而且这个程序是一个由基集合{序列，IF-THEN-ELSE，WHILE-DO }产生的结构程序。从而，定理得证。

2 - 24

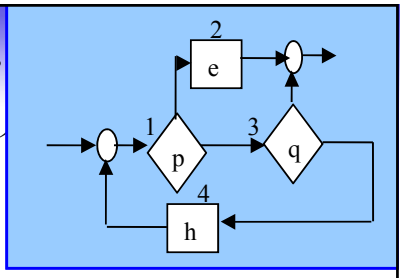
实例…

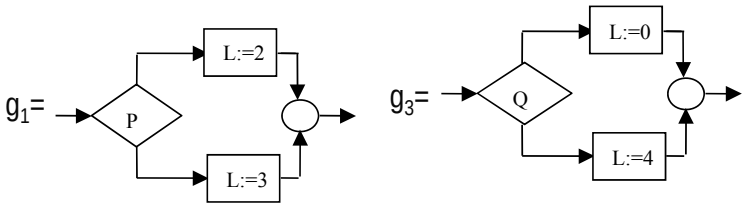
- 原程序
  - 编码
    - 规则1：给程序的函数结点、谓词结点编号，汇点略过
    - 规则2：出口线用它后面的结点的号码进行编号
    - 规则3：出口线后无结点，编号=0

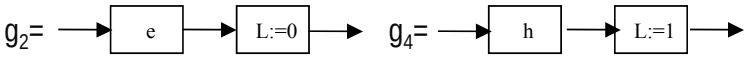


…实例…

- 编码后，对每一个编号构造一
  - 方法1：函数程序
  - 方法2：谓词程序

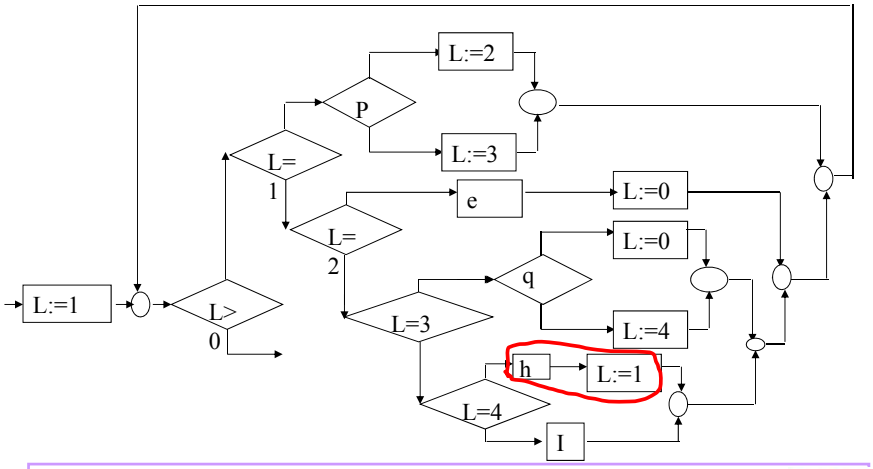






…实例

- 由g_i组合成新结构程序



问题：比较庞大，而且效率不高。

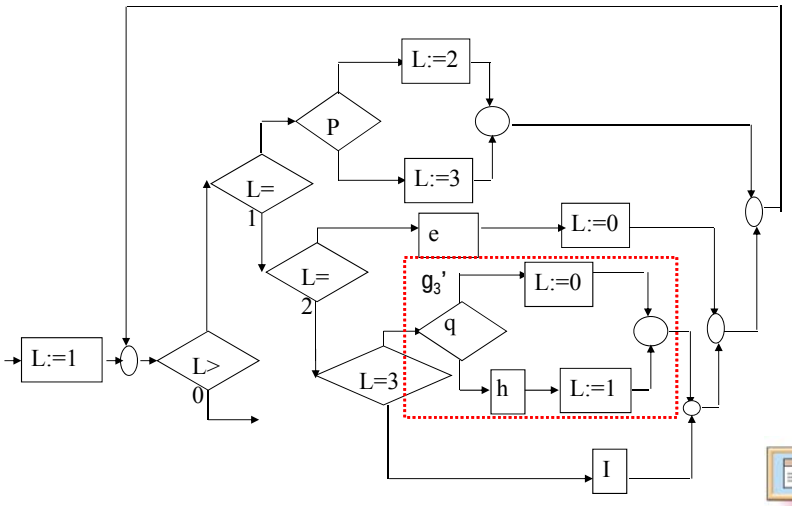
解决办法：需要简化，特别是消除一些多余的对L的测试与赋值。

…结构化定理

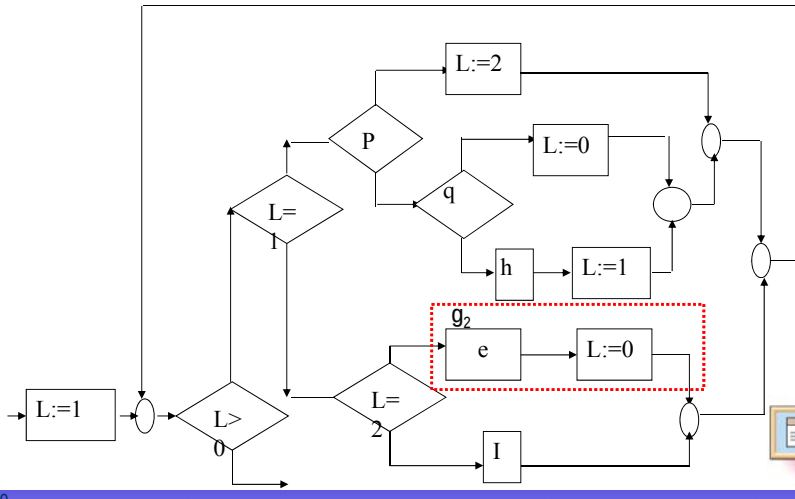
- 3） 递归结构程序
  - 基本思想：对于某些j>0，如果在g_j中不包含有赋值L:=j，那么可以用程序g_j代替所有的赋值L:=j。这样代替以后，由于j不再赋值给L，因而测试L=j可以从if-then-else结构中去掉。这个替换过程可以一直继续到下面两种情况出现为止：
    - a)除了 L:=0以外，所有的给L赋值均被消除；
    - b) 每一个余下的g_i'中都包含有相应的赋值L:=i。
      - g_i'是g_i经过替换以后得到的复合程序
  - 采用以上步骤得到的程序通常称为递归结构程序。

结构化→递归

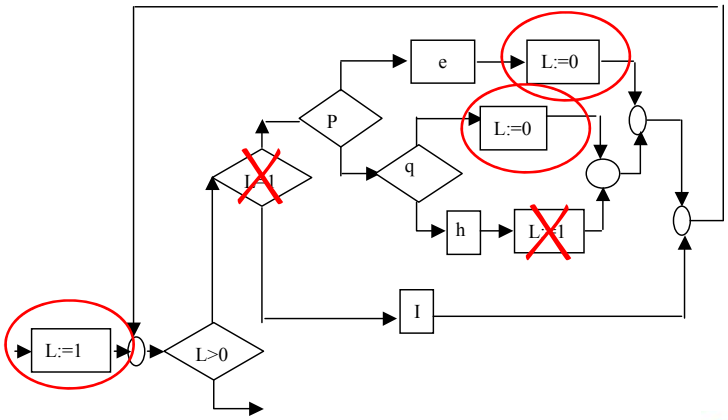
第一步，用g_4代替赋值L:=4，并且消除测试, 得到；



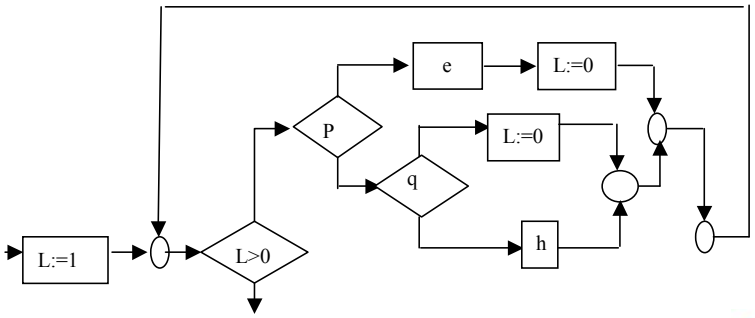
第二步，用代换后的g_3'，即L=3通路上的if-then-else程序代替赋值L:=3，并且消除测试L=3，得到



第三步，用g_2代替赋值L:=2，得到



- 第四步，L的初值为1，而且只有在程序的出口处才变为0，所以在循环中的赋值L:=1和测试L=1是不需要的，可以消除它们。

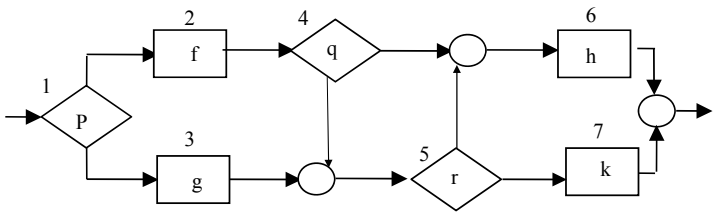


对应的程序

```
L:=1;
while L>0 do
Begin
  if  p  then
              begin
                  e;
                  L:=0;
              end;
         else
              begin
                 if q then
                          L:=0;
                      else
                          h;
               end;
End.
```

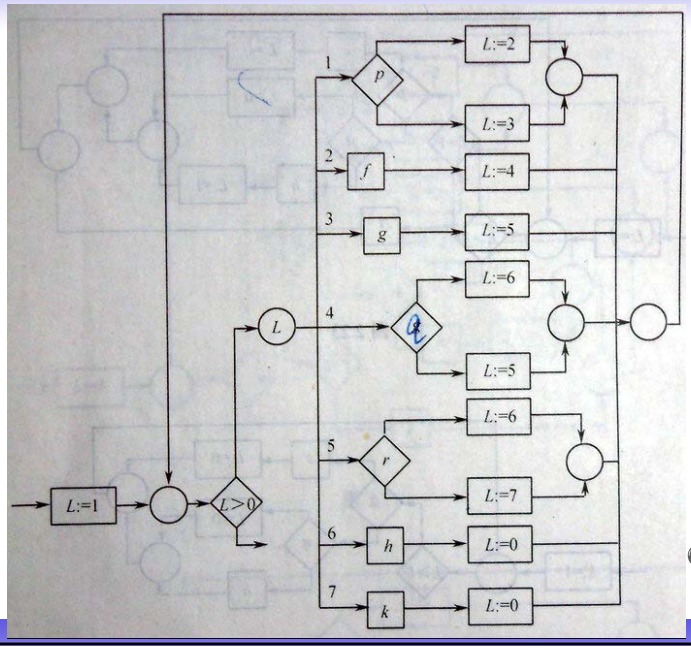
同步练习

- 将下列无循环程序转换为结构化程序



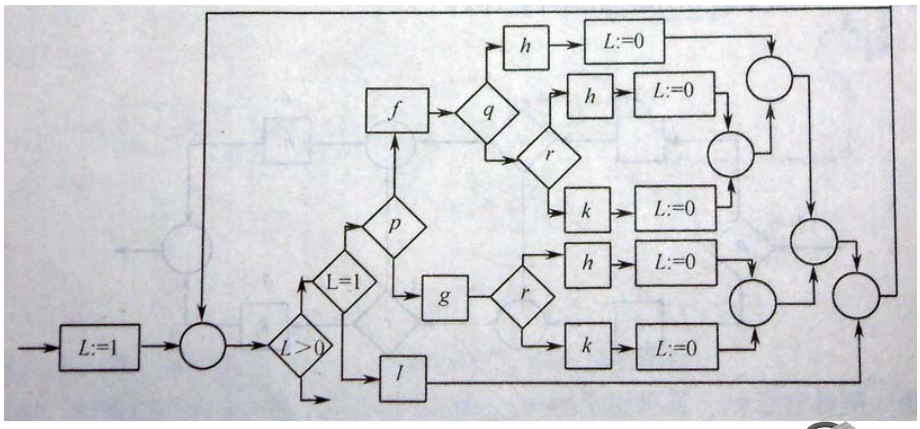
答案…

- 利用结构化定理中给出的方法，得到等价的结构化程序



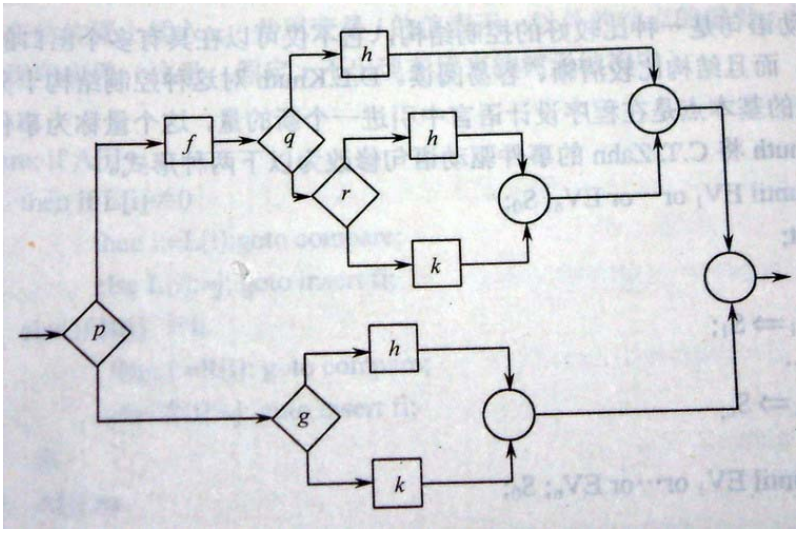
…答案…

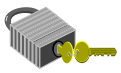
- 利用本章介绍方法进行逐步替换，可以得到递归结构程序



…答案

- 简化





内容线索

- 什么是结构化程序
- 结构化定理
- 一些新的控制结构
  - 事件驱动语句
  - 多重出口循环语句
  - 关于N+1/2循环问题
  - PDL语言的控制结构
  - 其他

事件驱动语句

until EV_1 or EV_2 or … or Ev_n do S_0
then case
  EV_1: S_1
  EV_2: S_2
  …
  EV_n: S_n

C.T.Zahn在1974年提出
含义：重复执行S_0,一直到出现事件EV_1，EV_2，…，EV_n之一为止。

多重出口循环语句

<出口语句>：：=exit <标号>
<循环语句>：：=<简单循环>|
<多重出口循环>
<多重出口循环>::=<简单循环>
<标号1>： S_1;
…
<标号n>： S_n;
endod: S_0;

G.V.Bochmann在1973年提出

关于N+1/2循环问题

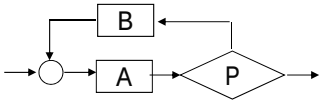
L1:A;

if p then goto L2 fi;

B;

goto L1;

L2:……



PDL语言的控制结构

序列	f;g
if-then:	if p then f fi
while-do:	while p do f od
do-until:	do f until p od
if-then-else:	if p then f else g fi
do-while-do:	do1 f while p do2 g od

其他一些控制结构…

- J.T.White与L.Presser提出的一种循环语句
  - loop
  - S1;
  - S2;
  - …
  - Sn;
  - end-loop
- 其中，S1; S2; …Sn中至少有一个包含下面两种跳出语句之一
  - exit（无条件跳出）
  - exit when p（条件跳出）

…其他一些控制结构…

- E.W.Dijkstra提出的一种推广的条件控制结构和循环控制结构
  - 条件结构
    - if
    - guard 1: action 1;
    - guard 2: action 2;
    - ……
    - guard n: action n;
    - fi
  - 循环结构
    - loop
    - guard 1: action 1;
    - guard 2: action 2;
    - ……
    - guard n: action n;
    - pool

…其他一些控制结构

- D.Gries提出的条件控制结构和循环结构
  - 条件
    - if
    - B1→S1
    - B2→S2
    - …….
    - Bn→Sn
    - fi;
  - 循环
    - do
    - B1→S1
    - B2→S2
    - …….
    - Bn→Sn
    - od;

作业1

- 1、证明结构化定理，并依据它将P38，1，4 转换为结构化程序。如果可能，请将其进一步转化为递归结构程序。
- 2、补充题：P38，2，3