

HỆ GP

NGUYỄN XUÂN HUY, BÙI QUANG NGỌC

I - ĐẶT VẤN ĐỀ

GP là một hệ thống phần mềm trợ giúp cho việc lập trình bằng lưu đồ (flowchart). Hệ thống có cơ chế giúp cho người lập trình soạn thảo các chương trình trên màn hình dưới dạng lưu đồ, lưu trữ lưu đồ trên đĩa từ, dịch lưu đồ chương trình sang ngôn ngữ lập trình thông dụng, thực hiện chương trình và sửa lỗi. Quá trình khai thác thử nghiệm cho thấy hệ thống giúp ích cho việc lập trình trực quan thông qua sự hỗ trợ của bàn phím và màn hình. Hệ thống GP phát huy tác dụng tốt với những người đang học lập trình và những lập trình viên hệ thống làm việc với các ngôn ngữ bậc thấp. Để đảm bảo được những nguyên lý của lập trình cấu trúc, hệ GP có cơ chế phát sinh chương trình con và ghép nối chúng thành các tờ hợp khác nhau. Ngoài ra, trong mỗi nút của lưu đồ, hệ cho phép dùng các cấu trúc lập như WHILE hoặc FOR Cơ chế gộp và tách nút của hệ thống cho phép người dùng tinh giản lưu đồ chương trình tới mức gọn nhất hoặc có thể dễ dàng sửa lại chương trình cho phù hợp với đòi hỏi bổ sung thêm chức năng khi thiết kế các hệ thống ứng dụng.

Ngôn ngữ của hệ GP đơn giản, để học đảm bảo tính cân đối, đối xứng của tập lệnh

II - CẤU TRÚC CỦA HỆ THỐNG

Hệ GP bao gồm 3 khối chức năng sau đây:

1. Khối trợ giúp soạn thảo chương trình bao gồm các chức năng:
 - Vẽ lưu đồ chương trình trên màn hình, cập nhật lưu đồ chương trình.
 - Soạn thảo nội dung vào các nút của lưu đồ chương trình.
2. Khối phục vụ việc lưu trữ và in ấn.

Mỗi chương trình do GP phát sinh được lưu trữ ở đĩa từ dưới dạng 3 loại tệp sau đây

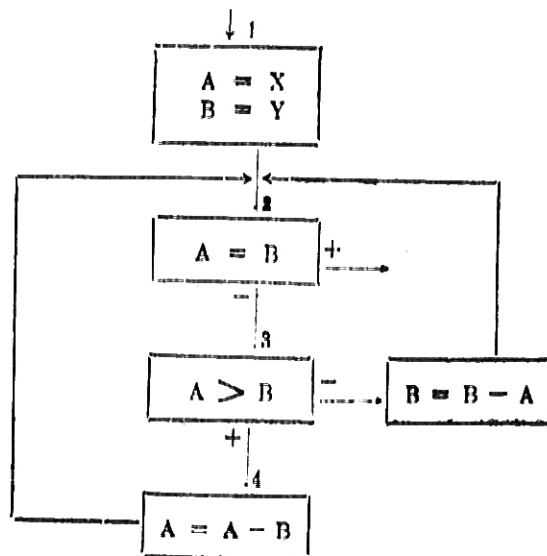
- Tệp *.GPG lưu trữ lưu đồ rỗng của chương trình.
- Tệp *.GPP lưu trữ nội dung các nút của lưu đồ chương trình.
- Các tệp *.BAS, *.PAS, *.ASL... lưu trữ chương trình do GP biên dịch từ lưu đồ khối sang dạng ngôn ngữ tương ứng BASIC, PASCAL, ASSEMBLER, v.v..

GP có thể cho hiện lưu đồ, nội dung của các nút của lưu đồ hoặc toàn bộ chương trình dịch (đi qua bước biên dịch) trên màn hình hoặc in ra giấy.

3. Khối phục vụ việc biên dịch từ lưu đồ sang một trong các ngôn ngữ lập trình nói trên.

III - HOẠT ĐỘNG CỦA HỆ THỐNG

1. Thí dụ minh họa. Để mô tả hoạt động của hệ GP chúng ta hãy xét bài toán tìm ước chung lớn nhất của 2 số nguyên dương X và Y. Thuật toán Euclide cho bài toán này được thể hiện qua lưu đồ dưới đây ($A = B = \text{UCLN}(X, Y)$).



Đồ lập trình cho bài toán này, người sử dụng thông qua hệ GP thực hiện 5 bước.

Bước 1 (Về lưu đồ). Thông qua các phím điều khiển, người dùng chuyển dịch con trỏ màn hình (cursor) tới vị trí mong muốn để bố trí các nút sau đó sẽ bấm lệnh tạo nút. Để về lưu đồ 1 (nhấn của nút đầu tiên được dùng để gắn nhãn cho lưu đồ) người lập trình sẽ thực hiện dãy thao tác sau:

- { Dịch chuyển con trỏ tới vị trí thích hợp cho nút 1 }
 - A1 { Lệnh tạo nút thao tác (loại A) có nhãn 1 }
 - { Dịch chuyển con trỏ tới vị trí thích hợp cho nút 2 }
 - B2,1 { Lệnh tạo nút điều kiện (loại B) có nhãn 2 sau đó nối một cung từ nút 1 tới nó - nút 2 }
 - { Dịch chuyển con trỏ... } B3, 2- { Lệnh tạo nút điều kiện có nhãn 3, nối cung từ nhánh phủ định của nút 2 (2-) tới nút vừa tạo }
 - { Dịch chuyển con trỏ... }
 - A4, 3
 - { Dịch chuyển con trỏ... }
 - A5, 3-
 - L4,2 { Lệnh lập một cung từ nút 4 tới nút 2 }
 - L5,2 { Lệnh lập một cung từ nút 5 tới nút 2 }
- Nhánh 2+ được coi là điểm kết thúc tự nhiên của lưu đồ
 Lưu đồ chương trình được tạo từ 3 loại nút và cung nối các nút.
 Nút loại A (nút thao tác) chứa các câu lệnh thao tác của chương trình.
 Nút loại B (nút điều kiện) chứa điều kiện cần kiểm tra trong lệnh rẽ nhánh. Nhánh dương (+) ứng với giá trị đúng, nhánh âm (-) ứng với giá trị sai của điều kiện.
 Nút loại C (nút gọi chương trình con).

Bước 2 (Viết nội dung vào các nút). Lệnh Wn chỉ thị cho hệ thống phục vụ cho người lập trình soạn thảo nội dung chương trình vào nút có nhãn n. Người dùng được cung cấp một tập lệnh soạn thảo tương tự như các lệnh soạn thảo văn bản thông thường, cụ thể là:

- Viết một dòng văn bản tiếp vào các dòng đã có,
- Xen 1 dòng văn bản vào giữa các dòng đã có,
- Xóa, sửa toàn bộ hoặc một phần văn bản (các dòng hoặc trên 1 dòng),
- Đánh phẩy lại các dòng theo bước 10,
- Hiện một số dòng văn bản.

Chú ý: Người lập trình có thể thực hiện xen kẽ các thao tác của bước 1 và bước 2. Thí dụ, có thể tạo các nút 1 và 5, viết nội dung vào các nút đó, sau đó tạo và viết nội dung của các nút còn lại rồi cuối cùng mới nối các cung.

Bước 3 (lưu trữ). Người lập trình có thể lưu trữ lưu đồ bằng lệnh S. Hệ GP sẽ phát sinh ra hai tệp (tệp *.GPG chứa lưu đồ chương trình và tệp *.GPP chứa nội dung chương trình. Dĩ nhiên, tệp GPP có thể là tệp trống nếu lệnh S được thực hiện ngay sau bước 1.

Để thử nghiệm, có thể bỏ qua bước lưu trữ đồ thực hiện ngay bước 4 dưới đây, khi không thấy sai sót mới gọi lệnh lưu trữ.

Bước 4 (hiển dịch) Lệnh P sẽ dịch lưu đồ chương trình đã soạn sang một trong những ngôn ngữ lập trình tùy theo chỉ thị của người dùng. Với chỉ thị B chẳng hạn, lưu đồ nói trên sẽ được dịch thành chương trình BASIC sau đây:

1 REM chương trình BASIC do GP phát sinh

2 GOTO 4

3 END

4 A = X : B = Y

5 REM kết thúc Nút 1, Khối 1

6 IF A = B THEN GOTO 3

7 REM kết thúc Nút 2, Khối 1

8 IF A > B THEN GOTO 13

9 REM Kết thúc Nút 3, Khối 1

10 B = B - A

11 GOTO 6

12 REM Kết thúc Nút 5, Khối 1

13 A = A - B

14 GOTO 6

15 REM Kết thúc Nút 4, Khối 1

16 REM Kết thúc Khối 1

Bước 5 (thực hiện). Bước thực hiện hiện nay vẫn nằm ngoài hệ GP, cụ thể là ở bước này người dùng phải kết thúc làm việc với GP bằng lệnh Q để sau đó gọi chương trình thông dịch BASIC và thực hiện chương trình được thu được ở bước 4.

IV - MỘT SỐ CHỨC NĂNG

Mục này giới thiệu một số chức năng tiện lợi của hệ GP nhằm nâng cao năng suất của người lập trình.

1. **Đổi nhãn nút:** Khả năng này giúp cho người lập trình đánh nhãn lại nút theo một cách tự mong muốn để sơ đồ trở thành dễ đọc hơn. Dĩ nhiên hệ GP đảm bảo cho lưu đồ vẫn giữ nguyên cấu trúc (tương quan giữa nút và cung) sau khi đã đổi lại nhãn của một số nút.

2. **Đổi kiểu nút:** Người lập trình có thể đổi một nút từ kiểu này sang kiểu khác trong số các kiểu A, B và C đã quy định. Khi đổi kiểu nút, các cung dương của nút B được hưởng quyền ưu tiên, cụ thể là khi đổi một nút n từ kiểu B sang kiểu A hoặc C thì cung đi từ nhánh âm của B sẽ bị xóa, cung đi từ nhánh dương vẫn được bảo lưu. Sau đó, nếu muốn thay đổi cung, người dùng sẽ sử dụng các lệnh đổi, lập hoặc xóa cung.

Lệnh đổi kiểu nút giúp cho người dùng sửa lại lưu đồ chương trình sau khi đã phát hiện ra nhầm lẫn.

3. **Xóa nút:** khi xóa một nút thì các cung đến nút đó và đi từ nút đó được tự động xóa theo.

4. **Xóa, sửa cung:** Khi đổi cung (x, y) sang cung (y, x) thì nhiều mũi tên được đảo lại, hệ thống sẽ tự động kiểm tra để đảm bảo cho mỗi nút chỉ có một cung đi đến nó. Khi đổi một cung (x, y) thành (x, z) thì cung cũ (x, y) sẽ tự động bị xóa để thay bằng cung mới (x, z).

5. **Gộp các nút:** Có thể gộp các nút loại A và C bằng lệnh I để sơ đồ được gọn và thoáng hơn. Khi gộp hai hay nhiều nút thì nút gộp (nút kết quả) sẽ mang nhãn của nút bị gộp đầu tiên. Các cung giữa các nút bị gộp dĩ nhiên sẽ tự mất đi, các cung đến nút bị gộp đầu tiên và cung đi ra từ nút bị gộp cuối cùng vẫn được bảo lưu và được gắn vào nút gộp.

6. Tách một nút: Có thể tách một nút loại A thành hai nút. Dùng nhiều phép tách liên tiếp ta có thể chia nhỏ một nút loại A thành nhiều nút cùng loại liên hoàn với nhau. Thao tác này đối xứng với thao tác gộp nút.

7. Tráo nội dung giữa hai nút cùng loại:

Ta có thể chuyển đổi nội dung của nút n sang nút m và ngược lại. Thao tác này giúp cho người lập trình tổ chức lại các nhóm lệnh cho hữu hiệu; thí dụ, đưa một số lệnh ra khỏi vòng lặp. Bằng cách sử dụng các lệnh sửa cung và đổi nhãn nút người dùng có thể tạo ra hiệu quả giống như thao tác tráo nội dung giữa hai nút.

8. Thực hiện chậm theo từng nút:

Nếu có chỉ thị này trước khi tiến dịch, hệ GP sẽ phát sinh ra một chương trình có khả năng thực hiện trong chế độ ngắt sau mỗi nút để người lập trình tự theo dõi và kiểm tra sản phẩm của mình.

9. Thu viện các khối: Người dùng có thể tạo ra từng chương trình con và gửi vào thư viện. Mỗi đơn vị chương trình con trong hệ GP được gọi là một khối. Sau khi kiểm tra hoạt động của từng khối như những chương trình độc lập, người dùng sẽ soạn một sơ đồ chương trình tổng hợp bao gồm các lệnh (nút loại C) gọi các khối như gọi một chương trình con.

10. Cơ chế giải thích, báo lỗi:

Hệ GP được định hướng cho việc truyền thụ kỹ thuật lập trình cho nên cơ chế giải thích và hướng dẫn được tổ chức tỉ mỉ.

Với những lệnh có tham trị, nếu người dùng quên chỉ định tham trị, hệ thống sẽ dùng chế độ hỏi đáp để bổ sung các tham trị cần thiết.

Ở mỗi khối chức năng hệ thống có một loại dấu nhắc để báo cho người dùng biết họ đang thao tác trong khối nào, thí dụ:

GP) cho biết đang ở mức đồng thể của hệ GP,

In) cho biết đang ở mức soạn thảo sơ đồ khối n,

WA) Cho biết đang viết lệnh vào nút thao tác (loại A) nhãn n...

Lệnh H ở bất kỳ mức nào giải thích cho người dùng cú pháp của lệnh thuộc mức đó và những lưu ý cần thiết trong tiến trình thao tác.

Lệnh ? Thông báo cáo đặc trưng của đối tượng được hỏi, thí dụ, khi gặp lệnh:

? 4, hệ GP sẽ thông báo:

Nút 4, loại A (Nút thao tác) và hiện trên màn hình nội dung của nút đó.

Việc thông báo lỗi được thực hiện kèm với hai tiếng còi ngắn. Dưới đây liệt kê vài lỗi thường gặp:

** Chưa có nút n

** Đã có nút n

** Không phải nút ĐIỀU KIỆN (B)

** Không phải nút THAO TÁC(A)

** Sai TÊN

** Sai số hiệu NÚT

** Không có CỬNG đi ra từ NÚT n

...

Chương trình dịch được ghi các chú thích và vị trí của từng nút trong từng khối nhằm giúp cho người dùng nhanh chóng phát hiện vị trí phát sinh lỗi trong sơ đồ.

11. Tối ưu hóa việc sinh mã : Hệ GP tự động khử các dãy GOTO liên hoàn, thí dụ, dãy lệnh

100 GOTO 102

101

102 GOTO 104

103

104 GOTO 110

Sẽ được thay bằng

100 GOTO 110

101 ...

Khi lưu trữ lưu độ trên đĩa từ hệ GP không lưu cả màn hình đồ thị mà chỉ lưu trữ tọa độ nút (2 giá trị : X và Y) và các cung (2 giá trị cho mỗi cung). Việc này làm giảm đáng kể miền nhớ ngoài dành cho chương trình.

V - TỒ CHỨC DỮ LIỆU

Mỗi nút của lưu đồ chương trình được mô tả qua kiểu sau :

TYPE Pnut = ^ Nut ;

Nut = RECORD

Số Hiệu Nút : Integer ;

Kiểu Nút : 1..3 ; {1 : A, 2 : B, 3 : C}

Tổng Số Dòng : Integer ;

Đầu Dòng : P dòng ;

XX, YY : Integer ; {Tọa độ Nút}

Left, Right : P nut ; {Nhánh +, -}

Ln, Rn : Integer ;

END ;

Pdòng = ^ Dòng ; {Dòng lệnh}

Dòng = RECORD

Số Hiệu Dòng : Integer ;

Nội Dung Dòng : String [125] ;

Next : Pdòng

END

Như vậy, mỗi lưu đồ được biểu diễn thành một đồ thị có hướng. Mỗi đỉnh của đồ thị có nhiều nhất hai cung đi ra.

Các dòng lệnh trong một nút được biểu diễn dưới dạng một danh sách, đầu của danh sách này được gắn vào trường Đầu dòng của NÚT tương ứng.

Tập *.GPG có kiểu

FGPG = FILE OF Nut

Tập *.GPP có kiểu

FGPP = FILE OF Dòng

Tập *.BAS có kiểu

FBAS = TEXT

Nói chung, Tập TEXT được dùng chung cho các chương trình đích.

Khi chuyển các Nút vào Tập *.GPG, hai trường Ln và Rn sẽ được sử dụng để ghi vị trí của bản ghi (số hiệu nguyên) của 2 nhánh dương và âm do Nút trở tới. Một ảnh xạ 1-1 sẽ được lập cho các cặp <left, Right> và <Ln, Rn>.

VI. HƯỚNG PHÁT TRIỂN

1. Trước mắt cần tạo ra một môi trường thống nhất cho hệ GP nhằm đảm bảo tính liên hoàn trong các bước thực hiện. Thí dụ, sau khi hệ GP phát sinh một chương trình BASIC, người dùng có thể thực hiện ngay chương trình đó trong GP mà không phải thông qua hai thao tác thủ công là :

- Ra khỏi hệ GP
- Gọi BASIC

2. Song song với nhiệm vụ 1, cần cải tiến hệ thống để đảm nhận được chức năng lắp ráp chương trình từ các đơn thể công nghệ theo các chỉ tiêu chất lượng định trước. Giải quyết được nhiệm vụ này, hệ GP sẽ trở thành một công cụ tốt để trang bị cho các điểm làm việc (Working station) của lập trình viên.

VII - MỘT VÀI THÔNG SỐ CỦA HỆ GP.

Ngôn ngữ cài đặt	:	PASCAL TURBO 3.1
Hệ điều hành	:	PC DOS, MS DOS
Dung lượng	:	30K (3000 dòng lệnh)
Số ổ đĩa mềm	:	1

Nhận ngày 21-9-1987

ABSTRACT

GP SYSTEM

GP is designed for processing program flowcharts. The main functions of system are as follows :

- Create and update program flowcharts on computer display.
- Generation of codes in BASIC, FORTRAN, ASSEMBLER, etc.

The GP conversational mode ensures easy operations for beginners-programmers and system programmers who are provided with easy - to - handle means of input, supervision and memorizing of graphics and data on floppy disks, reviewing this information on the display screen and correcting it whenever necessary.

The system programs are written in PASCAL TURBO 3.1 and run on compatible IBM PC computers with Operating systems PC DOS/MS DOS.