

R CÔNG NGHỆ VÀ TỰ ĐỘNG HÓA LẬP TRÌNH

(Báo cáo tại Hội thảo « Công nghệ phần mềm 87 »
 được tổ chức tại Bãi Cháy từ 24 đến 28/8/1987)

TÔ TUẤN

Trung tâm Toán - Máy tính

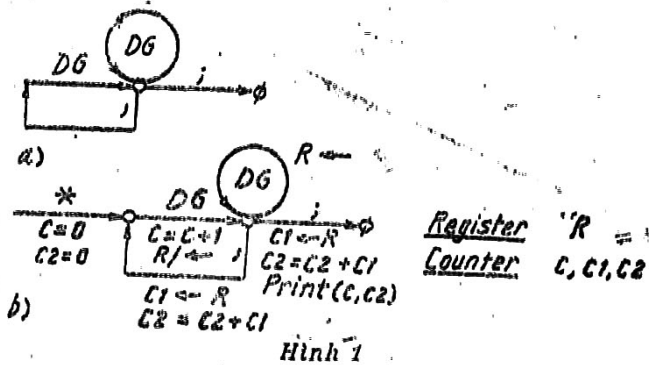
Vấn đề tự động hóa lập trình và nâng cao năng suất lao động của người lập trình về cơ bản được giải quyết theo hai hướng: làm ra ngôn ngữ lập trình tốt và xây dựng công nghệ lập trình, nghĩa là hình thành các phương pháp và công cụ quy định vấn đề lập trình cao. Vào giữa những năm 70 các chuyên gia của Viện điều khiển học Kiev dưới sự lãnh đạo của Viện sĩ V. M. Glusheop đã đề xuất ra cách tiếp cận thứ ba. Tư tưởng chính là cần phải xem xét lại những nguyên tắc làm việc cơ bản của máy tính. Phân tích những vấn đề của lập trình cho thấy là những khó khăn của lập trình nảy sinh không chỉ do sử dụng ngôn ngữ lập trình (đi hay công nghệ lập trình không hoàn chỉnh mà còn do những nguyên tắc xử lý tin đã quá cũ, không thích hợp với đòi hỏi của xử lý tin hiện đại (số lượng lớn, độ phức tạp về mặt cấu trúc cao). Nhược điểm chính trong nguyên tắc hoạt động của các máy tính hiện nay, là giữa dữ liệu và chương trình không có mối quan hệ trực tiếp (mối quan hệ gián tiếp qua người lập trình). Khi số lượng mối quan hệ này lớn vượt quá khả năng nhớ của người lập trình sẽ nảy ra những khó khăn mà phải vượt qua với chi phí lớn và với độ tin cậy của sản phẩm làm ra không cao.

R công nghệ lập trình ra đời dựa trên nguyên tắc làm việc mới của cái gọi là máy tính R [1]. Dữ liệu vào được nhập vào thiết bị điều khiển của máy tính R theo từng ký tự (trong máy tính truyền thống, dữ liệu được nhập vào bộ nhớ của máy), Chương trình của máy tính R quy định thứ tự nhập của các ký tự vào và thực hiện các thao tác tương ứng. Khi cần thiết máy tính R chuyển sang trạng thái khắc phục lỗi trong dữ liệu vào. Đặc trưng của quá trình lập trình cho máy tính R là sơ đồ logic của chương trình được đồng nhất với cấu trúc logic của dữ liệu vào, còn chức năng xử lý được tách rời. Như vậy để lập trình theo R công nghệ trước hết cần tập trung vào mô tả cấu trúc logic của dữ liệu vào không quan tâm đến các thao tác xử lý. Việc mô tả này được thực hiện bằng ngôn ngữ đồ thị định hướng có trọng số. Trên các cung có thể sử dụng các ký tự của dữ liệu vào, các tần tử phản ánh tính chất này hay tính chất kia của dữ liệu cần xử lý. Bước tiếp theo của lập trình là hoàn thiện mô tả nhận được ở bước trước bằng các thao tác cần phải thực hiện. Các thao tác này được đặt ở dưới các cung của đồ thị và được thực hiện mỗi khi tần tử tương ứng nhận giá trị đúng hoặc khi ký tự hiện gặp trùng với ký tự đặt trên cung. Lập trình theo R công nghệ được minh họa trên ví dụ xử lý danh sách các số nguyên cách nhau bởi dấu phẩy và kết thúc bằng dấu chấm phẩy. Cần phải tính tổng các số nguyên và số các số nguyên trong danh sách.

a) Bước 1, (mô tả cấu trúc logic của dữ liệu vào). Dữ liệu ban đầu là một xâu ký tự bao gồm các chữ số hệ 10, các dấu phẩy và kết thúc bằng dấu chấm phẩy. Cấu trúc logic của nó có thể được mô tả trên ngôn ngữ đồ thị định hướng có trọng số (còn gọi là ngôn ngữ R) như trên hình 1.a.

Để thấy rằng ngôn ngữ R rất giống với ngôn ngữ lược đồ cú pháp của N. Wirth [2]. Sự khác biệt ở chỗ là các ký tự có thể được sử dụng trong xâu vào được đặt ở trên các cung. Ký tự với tên DG được gọi là ký tự đồng hợp (Synterm) với định nghĩa trên BNF ở dạng:

DG ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9



b) Bước 2 (hoàn thiện mô tả cấu trúc logic của dữ liệu vào bằng cách thêm vào dưới các cung những thao tác cần phải thực hiện). Các thao tác đặt dưới các cung là các thao tác trên các bộ nhớ trừu tượng của máy tính R. Trong ví dụ này cần dùng đến 4 bộ nhớ trừu tượng: Một bộ nhớ kiểu thanh ghi R và ba bộ nhớ kiểu đếm c, c1, c2. Bộ nhớ kiểu thanh ghi (Register) giống với biến kiểu LSTRING trong PASCAL, MS/DOS hoặc biến CHARVAR trong PL/1, Còn

bộ nhớ đếm (Counter) tương tự như biến kiểu INTEGER của PASCAL. Đồ thị trên hình 1.a sau khi hoàn thiện có dạng như trên hình 1.b. Đồ thị 1.b so với đồ thị 1.a được mở rộng thêm một cung đánh dấu *, nghĩa là hai thao tác tương ứng $c = 0$ và $c2 = 0$ luôn được thực hiện (c dùng để đếm số các số nguyên trong danh sách, c2 dùng để tích lũy tổng các số nguyên). Thao tác R/\leftarrow có nghĩa là R đầu tiên được làm sạch (có độ dài bằng 0), sau đó ký tự hiện gặp được nạp vào và trở thành ký tự đầu tiên (chữ số đầu tiên của số nguyên hiện thời); Thao tác $R\leftarrow$ chỉ nạp ký tự hiện gặp vào bộ nhớ R. Mỗi khi gặp dấu phẩy (kết thúc một số nguyên và chuyển sang số nguyên tiếp sau) cần biến đổi nội dung của R sang dạng có thể thao tác được bằng các phép toán số học. Để làm điều này cần thực hiện $c1 \leftarrow R$, trong đó c1 - bộ nhớ đếm. Kết quả được in ra khi gặp dấu chấm phẩy, dấu kết thúc của xâu ký tự vào.

c) Bước 3 (mã hóa đồ thị định hướng có trọng số ở dạng tuyến tính), Bước này được thực hiện để nhập chương trình viết xong ở bước 2 vào máy tính. Để mã hóa cần gán cho mỗi nút của đồ thị một tên bất kỳ, nút cuối (ϕ) có tên đặc biệt là EXIT. Trong dạng tuyến tính đồ thị 1.b bây giờ trở thành:

	R - program	XLYSONGUYEN	
	Register	R	
	Counter	c, c1, c2	
BEG	*	c = 0, c2 = 0	LOOP
LOOP	DG	c = c + 1; R/\leftarrow	NEXT
NEXT	DG	R\leftarrow	NEXT
		c1 ← R; c2 = c2 + C1	LOOP
		c1 ← R; c2 = c2 + c1;	
		PRINT (c, c2)	EXIT
	END		

Có thể thử nghiệm chương trình XLYSONGUYEN kể trên với các số liệu sau:

1268, 275, 112689;

7, 51, 3689;

Thống kê cho thấy R chương trình (dạng tuyến tính của đồ thị định hướng có trọng số) ngắn hơn (bao gồm ít ký tự hơn) so với các chương trình tương đương trên các ngôn ngữ lập trình truyền thống (PASCAL, ADA...) từ 1,5 đến 3 lần [1]. Để tự động hóa lập trình theo R công nghệ trên máy tính họ ES (máy tính R được mô hình hóa trên ES) Viện điều khiển học Kiev đã xây dựng thành công tổ hợp công nghệ với tên RTK OS/ES. Lĩnh vực áp dụng có hiệu quả hơn cả của RTK là các bài toán xử lý thông tin dạng văn bản (xâu ký tự), trong số đó là các bài toán xây dựng chương trình dịch cho các ngôn ngữ đa năng hoặc chuyên dụng, xây dựng các hệ tự động hóa thiết kế, cài đặt các ngôn ngữ giao tiếp với cơ sở dữ liệu... Từ đầu những năm 80 đã cài đặt R công nghệ lập trình không dùng giấy tức là quá trình thiết kế và cài đặt hoàn toàn chỉ dùng công cụ nhất quán và xuyên suốt là đồ thị định hướng có trọng số (đồ thị R) và được vẽ trực tiếp trên màn hình của Display. Tổ hợp RTK quản lý các đồ thị này không qua giai đoạn mã hóa ở dạng tuyến tính [3].

Chương trình trên ngôn ngữ đồ thị R có tính chất đặc biệt là có thể làm việc trong chế độ sinh ra những dữ liệu mà nó có thể xử lý. Quá trình sinh như vậy có thể được thực hiện với sự giúp đỡ của bộ sinh số ngẫu nhiên. Đối với đồ thị trên hình 1.a từ nút thứ nhất chỉ xuất phát có một cung và do đó di chuyển đến nút thứ hai bao giờ cũng duy nhất. Trên cung có ghi DG đó đó có thể có 10 khả năng, giả sử $G(10)=3$, trong đó G—bộ sinh số ngẫu nhiên, như vậy sẽ sinh ra ký tự 2. Từ nút thứ 2 có ba cung xuất phát và giả sử $G(3) = 1$ ở thời điểm này, nghĩa là chọn được cung DG. Theo cung nhận được $G(10)$ có thể cho số 5 và kết quả sinh được 24... Như vậy trong chế độ sinh có thể sinh ra xâu ký tự theo quy tắc giả ngẫu nhiên. Xâu vừa được sinh ra có thể được dùng làm lối vào cho chương trình trong chương trình với đồ thị 1.b. Nói chung quy trình thử nghiệm (testing) có thể được thực hiện như sau. Song song với thiết kế chương trình theo R công nghệ, một chương trình thử nghiệm được xây dựng. Chương trình này sẽ làm việc trong chế độ sinh để tạo ra các số liệu thử nghiệm cho chương trình thử nhất. Trong khuôn khổ của RTK một môi trường thử nghiệm được xây dựng với cơ sở là hai máy tính R: Máy tính sinh và máy tính kiểm soát. Lối ra của máy tính sinh sẽ là lối vào của máy tính kiểm soát. Công việc của hai máy được tổ chức sao cho trong quá trình sinh không hạn chế các số liệu thử nghiệm (test) lưu trữ lại tất cả lịch sử của quá trình như số lần chạy đúng, số lần chạy sai, số liệu đúng, số liệu sai, nơi có lỗi... Theo [1] độ phức tạp thử nghiệm trong R công nghệ giảm đi rất nhiều so với cách làm truyền thống. Thông thường độ phức tạp tăng lên theo quy luật hàm mũ từ thử nghiệm độc lập đến thử nghiệm toàn bộ. Thử nghiệm theo R công nghệ chỉ đòi hỏi độ phức tạp tuyến tính. Nhược điểm của cách thử nghiệm theo R công nghệ ở chỗ là cần phải thiết kế hai chương trình. Trên thực tế điều này đòi hỏi chi phí phụ nhất định. Theo cách tiếp cận đề xuất trong [4] có thể không cần phải viết chương trình sinh số liệu thử nghiệm mà sử dụng ngay chính chương trình cần thử nghiệm làm chương trình vừa chạy trong chế độ sinh vừa chạy trong chế độ thử nghiệm (trong ví dụ đã xét dùng ngay chính chương trình 1.b để chạy trong chế độ vừa sinh vừa thử nghiệm). Trong chế độ liên hợp này sẽ không dùng tới số liệu vào chuẩn bị bằng người lập trình hay sinh ra bởi chương trình khác. Số liệu vào sẽ được lấy từ chính quá trình sinh theo chương trình cần thử nghiệm. Cách tiếp cận này nhằm khi giải phóng người lập trình khỏi công việc không lấy gì làm thích thú như chuẩn bị ra các số liệu thử nghiệm hoặc viết thêm chương trình thử hai làm nhiệm vụ sinh ra các số liệu giả ngẫu nhiên. Chế độ liên hợp cho phép phát hiện nhanh nhiều lỗi mà người dùng phạm phải, nhất là các lỗi trong các thao tác trên bộ nhớ trừu tượng đặt ở dưới các cung của chương trình trên ngôn ngữ đồ thị R. Để minh họa cho cách thử nghiệm liên hợp ta xét chương trình XLYSO-NGUYEN đã viết dưới dạng tuyến tính. Ở đầu chương trình, từ nút BEG xuất phát một cung duy nhất, do đó cung này bao giờ cũng được chọn. Như vậy hai thao tác làm sạch bộ nhớ đếm c và c2 được thực hiện. Nút tiếp sau là nút LOOP cũng có một cung xuất phát duy nhất (cung DG). Giả sử ở thời điểm này $G(10)=3$ và như vậy ký tự được chọn sẽ là 2. Bộ đếm c được tăng lên 1 theo thao tác $c=c+1$ và R được nạp ký tự 2 theo R/\leftarrow . Bây giờ trạng thái chương trình chuyển sang nút NEXT mà từ đó xuất phát 3 cung. Giả sử $G(3)=1$ nghĩa là cung DG được chọn. Bộ sinh $G(10)$ có thể sinh ra số thứ tự của một ký tự số nào đó, ví dụ số 5 nghĩa là ký tự 4 được sinh ra. Sau khi thực hiện thao tác $R \leftarrow$ nội dung của R sẽ là xâu 24 và nút tiếp theo vẫn là chính nút NEXT. Vào thời điểm này cung với dấu phẩy ở trên có thể được chọn. Nội dung của R sẽ được biến đổi từ dạng xâu ký tự sang dạng có thể gán cho bộ nhớ đếm c1. Tích lũy tổng được thực hiện bởi $c2=c2+c1$. Sau đó trạng thái của chương trình chuyển sang LOOP và số nguyên tiếp theo phải được sản sinh ra. Cứ như vậy cho đến khi ở trạng thái NEXT chọn được cung với dấu chấm phẩy. Khi đó số nguyên cuối cùng trong danh sách được xử lý và kết quả chứa trong c và c2 được in ra. Trong quá trình thử nghiệm theo chế độ liên hợp có thể xảy ra ngắt chương trình do lỗi lập trình. Vị trí ngắt, xâu ký tự được sản sinh dở dang và trạng thái các bộ nhớ trừu tượng cho phép dễ dàng xác định được vị trí lỗi cùng với nguyên nhân của nó. Nếu thử nghiệm chạy trọn vẹn đến cùng, trên cơ sở phân tích xâu ký tự được sản sinh và kết quả in ra có thể xác định được tình đúng đắn của quá trình vừa thực hiện (tức là tình đúng đắn của chương trình đối với số liệu sinh ra tương ứng). Việc thử nghiệm có thể được tiếp tục lặp lại theo yêu cầu của người dùng (có thể chỉ ra số lần cần thực hiện). Khi đó trong chế độ giả ngẫu nhiên một xâu ký tự khác lại được sản sinh và kết quả tương ứng được in ra.

Tổ hợp tự động lập trình theo R công nghệ ngoài họ máy ES còn được cài đặt trên họ máy mini SM, trên các máy tính điều khiển loại KLETRONICA-60 và các máy tính cá nhân DVK-2. Kinh nghiệm sử dụng R công nghệ cho thấy hiệu suất lập trình được nâng cao rõ rệt so với lập trình truyền thống. Chẳng hạn một hệ thống bao gồm 60 nghìn lệnh được cài đặt với hiệu suất tối đa là 100 lệnh một người một ngày. Trong khi đó theo công nghệ thông thường con số tương ứng là 5-8 lệnh [1]. Để nắm vững R công nghệ chỉ cần từ 1-2 tuần đến 1-2 tháng. Ở Việt Nam tư tưởng của R công nghệ đã được sử dụng để cài đặt hệ quản trị cơ sở dữ liệu theo mô hình quan hệ DB1 [5] và hệ in bằng biểu tự động theo APS hỗ trợ cho hệ DBASE III trên máy tính cá nhân IBMPC. Tổ hợp RTK OS/ES hiện có tại trung tâm Toán-Máy tính Hà Nội.

Nhận ngày 19-9-1987

TÀI LIỆU THAM KHẢO

1. Вельбицкий И. В., и другие, Технологический комплекс производства программ на машинах ЕС ЭВМ, БЭСМ 6, М., Статистика, 1980.
2. Wirth N., Cấu trúc dữ liệu+giải thuật=chương trình, Hà Nội, NXB Thống kê, 1982.
3. Вельбицкий И.В., Безбумажная Р-технология программирования широкого применения—Вычислительная техника социалистических стран, выпуск 18, 1985.
4. Tô Tuấn, Về hình thành và phát triển R-công nghệ lập trình. Khoa học kỹ thuật và tự động hóa chỉ huy, N^o-3, 1982, 28-52.
5. Tô Tuấn, Các vấn đề xây dựng ngân hàng dữ liệu tự động hóa. Khoa học kỹ thuật và tự động hóa chỉ huy, N^o-1, 1984, 59-77.

РЕЗЮМЕ

R—Технология и автоматизация программирования
Излагается основная идея R-технологии программирования. Обсуждается связь R-технологии с проблемой автоматизации программирования. Предлагается новый подход к тестированию R-программ.