Phương pháp biểu diễn thuật toán

Có 3 phương pháp biểu diễn thuật toán:

1. Ngôn ngữ tự nhiên

Trong cách biểu diễn thuật toán theo ngôn ngữ tự nhiên, người ta sử dụng ngôn ngữ thường ngày để liệt kê các bước của thuật toán (Các ví dụ về thuật toán trong mục 1 của chương sử dụng ngôn ngữ tự nhiên). Phương pháp biểu diễn này *không* yêu cầu người viết thuật toán cũng như người đọc thuật toán phải nắm các quy tắc. Tuy vậy, cách biểu diễn này thường dài dòng, không thể hiện rõ cấu trúc của thuật toán, đôi lúc gây hiểu lầm hoặc khó hiểu cho người đọc. Gần như không có một quy tắc cố định nào trong việc thể hiện thuật toán bằng ngôn ngữ tự nhiên. Tuy vậy, để dễ đọc, ta nên viết các bước con lùi vào bên phải và đánh số bước theo quy tắc phân cấp như 1, 1.1, 1.1.1...

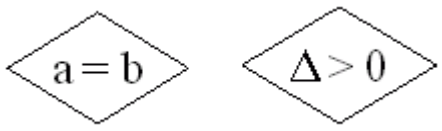
2. Lưu đồ - sơ đồ khối

Lưu đồ hay sơ đồ khối là một công cụ trực quan để diễn đạt các thuật toán. Biểu diễn thuật toán bằng lưu đồ sẽ giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán. Phương pháp lưu đồ thường được dùng trong những thuật toán có tính rắc rối, khó theo dõi được quá trình xử lý.

Để biểu diễn thuật toán theo sơ đồ khối, ta phải phân biệt hai loại thao tác. Một thao tác là thao tác chọn lựa dựa theo một điều kiện nào đó. Chẳng hạn: thao tác "*nếu a = b thì thực hiện thao tác B2, ngược lại thực hiện B4*" là thao tác chọn lựa. Các thao tác còn lại không thuộc loại chọn lựa được xếp vào loại *hành động*. Chẳng hạn, "*Chọn một hộp bất kỳ và để lên đĩa cân còn trống.*" là một thao tác thuộc loại hành động.

2.1. Thao tác chọn lựa (decision)

Thao tác chọn lựa được biểu diễn bằng một hình thoi, bên trong chứa biểu thức điều kiện.



2.2. Thao tác xử lý (process)

Thao tác xử lý được biểu diễn bằng một hình chữ nhật, bên trong chứa nội dung xử lý.

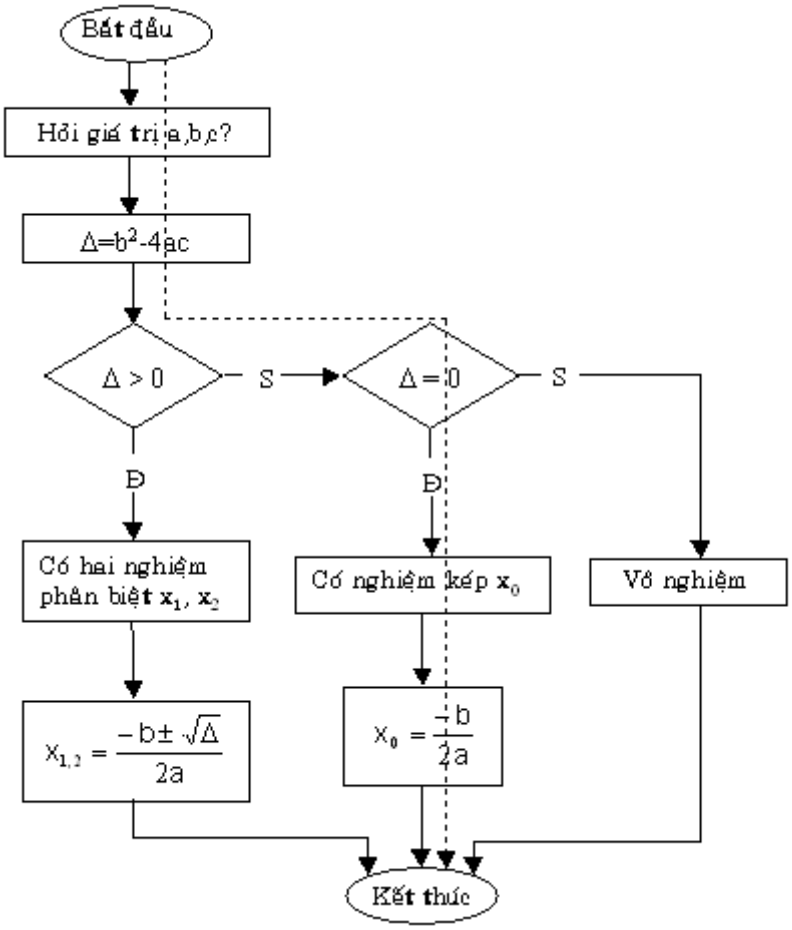
2.3. Đường đi (route)

Khi dùng ngôn ngữ tự nhiên, ta mặc định hiểu rằng quá trình thực hiện sẽ lần lượt đi từ bước trước đến bước sau (trừ khi có yêu cầu nhảy sang bước khác). Trong ngôn ngữ lưu đồ, do thể hiện các bước bằng hình vẽ và có thể đặt các hình vẽ này ở vị trí bất kỳ nên ta phải có phương pháp để thể hiện trình tự thực hiện các thao tác.

Hai bước kế tiếp nhau được nối bằng một cung, trên cung có mũi tên để chỉ hướng thực hiện. Chẳng hạn trong hình dưới, trình tự thực hiện sẽ là B1, B2, B3.

Từ thao tác chọn lựa có thể có đến hai hướng đi, một hướng ứng với điều kiện thỏa và một hướng ứng với điều kiện không thỏa. Do vậy, ta dùng hai cung xuất phát từ các đỉnh hình thoi, trên mỗi cung có ký hiệu Đ/Đúng/Y/Yes để chỉ hướng đi ứng với điều kiện thỏa và ký hiệu S/Sai/N/No để chỉ hướng đi ứng với điều kiện không thỏa.

Ví dụ: Lưu đồ thuật toán giải phương trình bậc 2

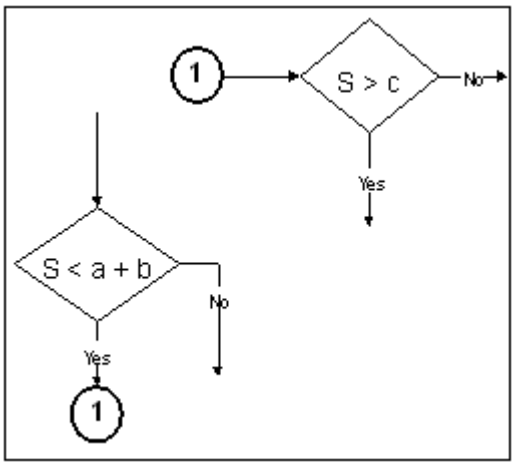


2.4. Điểm cuối (terminator)

Điểm cuối là điểm khởi đầu và kết thúc của thuật toán, được biểu diễn bằng hình ovan, bên trong có ghi chữ bắt đầu/start/begin hoặc kết thúc/end. Điểm cuối chỉ có cung đi ra (điểm khởi đầu) hoặc cung đi vào (điểm kết thúc). Xem lưu đồ thuật toán giải phương trình bậc hai ở trên để thấy cách sử dụng của điểm cuối.

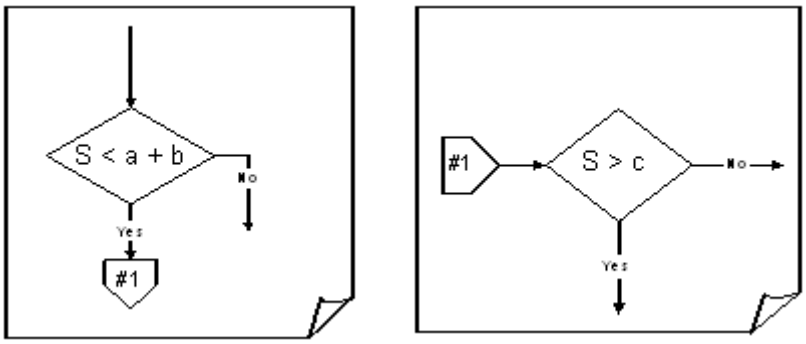
2.5. Điểm nối (connector)

Điểm nối được dùng để nối các phần khác nhau của một lưu đồ lại với nhau. Bên trong điểm nối, ta đặt một ký hiệu để biết sự liên hệ giữa các điểm nối.



2.6. Điểm nối sang trang (off-page connector)

Tương tự như điểm nối, nhưng điểm nối sang trang được dùng khi lưu đồ quá lớn, phải vẽ trên nhiều trang. Bên trong điểm nối sang trang ta cũng đặt một ký hiệu để biết được sự liên hệ giữa điểm nối của các trang.



Ở trên chỉ là các ký hiệu cơ bản và thường được dùng nhất. Trong thực tế, lưu đồ còn có nhiều ký hiệu khác nhưng thường chỉ dùng trong những lưu đồ lớn và phức tạp. Đối với các thuật toán trong cuốn sách này, ta chỉ cần sử dụng các ký hiệu trên là đủ.

3. Mã giả

Tuy sơ đồ khối thể hiện rõ quá trình xử lý và sự phân cấp các trường hợp của thuật toán nhưng lại cồng kềnh. Để mô tả một thuật toán nhỏ ta phải dùng một không gian rất lớn. Hơn nữa, lưu đồ chỉ

phân biệt hai thao tác là rẽ nhánh (chọn lựa có điều kiện) và xử lý mà trong thực tế, các thuật toán còn có thêm các thao tác lặp.

Khi thể hiện thuật toán bằng mã giả, ta sẽ *vay mượn* các cú pháp của một ngôn ngữ lập trình nào đó để thể hiện thuật toán. Tất nhiên, mọi ngôn ngữ lập trình đều có những thao tác cơ bản là xử lý, rẽ nhánh và lặp. Dùng mã giả vừa tận dụng được các khái niệm trong ngôn ngữ lập trình, vừa giúp người cài đặt dễ dàng nắm bắt nội dung thuật toán. Tất nhiên là trong mã giả ta vẫn dùng một phần ngôn ngữ tự nhiên.

Ví dụ:

```
if Delta > 0 then begin
□x1=(-b-sqrt(delta))/(2*a)
□x2=(-b+sqrt(delta))/(2*a)
□xu□t k□t qu□: phương trình có hai nghiệm là x1 và x2
else if delta = 0 then
□xu□t k□t qu□: phương trình có nghiệm kép là -b/(2*a)
else {trường hợp delta < 0 }
□xu□t k□t qu□: phương trình vô nghiệm
```

Revision #3
Created 15 April 2020 16:28:21 by Laptrinh.vn
Updated 27 June 2023 09:12:35 by Laptrinh.vn