Chương 3

PHÂN RÃ BÀI TOÁN - TÌM KIẾM LỜI GIẢI TRÊN ĐỒ THỊ VÀ/ HOẶC

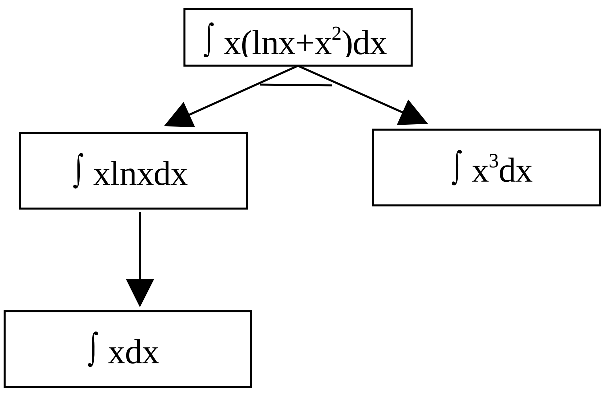
1. Đặt vấn đề.

Trong chương 2, chúng ta đã nghiên cứu việc biểu diễn bài toán thông qua các trạng thái và các toán tử. Khi đó việc tìm lời giải của bài toán được quy về việc tìm đường đi trong không gian trạng thái. Trong chương này chúng ta sẽ nghiên cứu một phương pháp luận khác để giải quyết vấn đề, dựa trên việc quy vấn đề về các vấn đề con.

Ý tưởng chủ yếu là xuất phát từ bài toán ban đầu, tách ra các bài toán con, quá trình này tiếp tục đối với các bài toán con cho đến khi gặp các bài toán sơ cấp (bài toán có lời giải ngay).

Ví dụ 1. Xét bài toán tính tích phân $\int x(\ln x + x^2)dx$.

Thông thường để tính tích phân bất định, chúng ta thường sử dụng các quy tắc tính tích phân: tích phân của tổng, quy tắc tích phân từng phần hay các phép biến đổi v.v… để đưa tích phân cần tính về tích phân của các hàm số sơ cấp mà chúng ta đã biết cách tính. Đối với tích phân trên, áp dụng quy tắc tích phân của tổng ta đưa về hai tích phân $\int$ xlnxdx và tích phân $\int x^3dx$. Áp dụng quy tắc tích phân từng phần ta đưa tích phân $\int$ xlnx về tích phân $\int$ xdx. Quá trình trên có thể biểu diễn bởi đồ thị trong Hình 1.

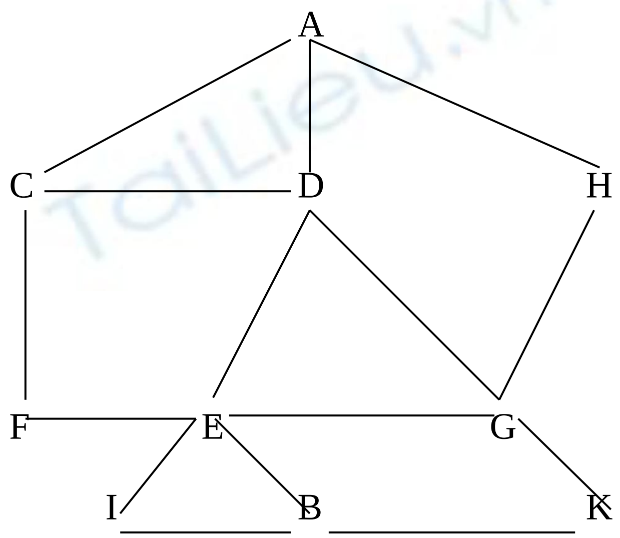


Hình 1.

Trong bài toán tích phân, các tích phân cơ bản là các trạng thái kết thúc.

Ví dụ 2. Bài toán tìm đường đi trên bản đồ giao thông.

Bài toán này đã được phát biểu như bài toán tìm đương đi trong không gian trạng thái, trong đó mỗi trạng thái ứng với một thành phố, mỗi toán tử ứng với một con đường, nối thành phố này với thành phố khác. Bây giờ ta đưa ra một cách bểu diễn khác dựa trên việc quy vấn đề về các vấn đề con.. Xét bản đồ giao thông giữa các thành phố trong Hình 2.



Hình 2.

Giả sử ta cần tìm đường đi từ thành phố A đến thành phố B. Có một con sông chảy qua hai thành phố E và G và có cầu qua sông ở mỗi thành phố đó. Như vậy mọi đường đi từ A đến B đều phải đi qua E hoặc G. Khi đó bài toán tìm đường đi từ A đến B được quy về một trong hai bài toán:

1) Bài toán tìm đường đi từ A đến B qua E

2) Bài toán tìm đường đi từ A đến B qua G

Mỗi một bài toán trên lại có thể phân nhỏ như sau:

1) Bài toán tìm đường đi từ A đến B qua E được quy về:

1.1. Tìm đường đi từ A đến E và

1.2. Tìm đường đi từ E đến B

2) Bài toán tòm đường đi từ A đến B qua G được quy về:

2.1. Tìm đường đi từ A đến G và

2.2. Tìm đường đi từ G đến B

Tổng quát, từ bài toán P ta đưa về một trong các trường hợp:

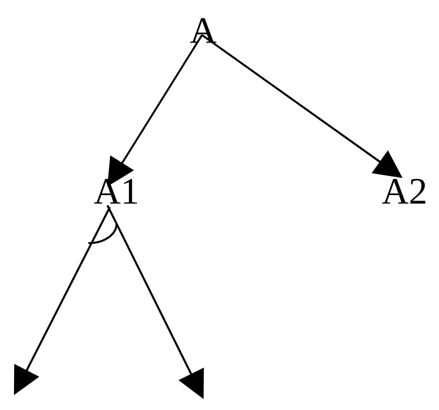
- Đưa P về các bài toán tương đương: P1, P2,..., Pk
- Đưa P về các bài toán con: P1, P2,..., Pk

Phương pháp phân chia bài toán ban đầu như trên đã gặp trong lập trình truyền thống với cách gọi “chia để trị” , “Modul hoá”.

2. Đồ thị VÀ/HOẶC:

Không gian trạng thái mô tả việc quy vấn đề về các vấn đề con có thể biểu diễn dưới dạng đồ thị định hướng đặc biệt gọi là đồ thị và/hoặc. Đồ thị này được xây dựng như sau:

Mỗi bài toán ứng với một đỉnh của đồ thị. Nếu có một toán tử quy bài toán về các bài toán tương đương thì sẽ có các cung đi từ bài toán xuất phát đến các bài toán tương đương đó. Nếu một toán tử quy bài toán về các bài toán con thì cũng có các cung nối từ bài toán xuất phát đến các bài toán con, ngoài ra giữa các cung này cũng có đường nới với nhau.. Chẳng hạn, giả sử bài toán A được đưa về hai bài táon tương đương A1 và A2. Bài toán A1 lại được quy về hai bài toán con B1 và B2, ta có biểu diễn như hình 3.



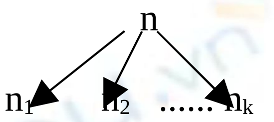
B1 B2

Hình 3

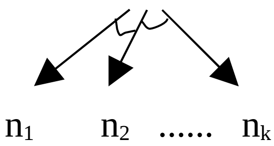
Một cách hình thức ta có thể định nghĩa đồ thị và/hoặc như sau:

Đồ thị G = (V, E) được gọi là đồ thị VÀ/HOẶC nếu $\forall n \in V$, T(n) hoặc các bài toán con của n (n gọi là các đỉnh VÀ) hoặc là tập các bài toán tương đương với n (n gọi là đỉnh HOẶC).

Cách biểu diễn như sau:



n được gọi là đỉnh HOẶC ($n \Leftrightarrow n_1 \Leftrightarrow ... \Leftrightarrow n_k$)



n được gọi là đỉnh VÀ ($n \Leftrightarrow n_1 \cap ... \cap n_k$)

Khi đó để giải bài toán n ta phải tìm đồ thị con của G là một cây có gốc là đỉnh xuất phát n sao cho mọi đỉnh trên đồ thị con này đưa về được các bài toán sơ cấp (đỉnh kết thúc).

Nhận xét: Gọi VA: tập các đỉnh VÀ

VO: tập các đỉnh HOẶC

- Nếu VA= $\varnothing \Rightarrow$ tìm lời giải trên đồ thị biểu diễn bằng không gian trạng thái

Khi đó:

- Bài toán n được gọi là giải được nếu:

+ hoặc n là đỉnh kết thúc

+ hoặc T(n)={n_1, n_2,..., n_k} và nếu n là đỉnh HOẶC $\Rightarrow \exists i \in (1...k)$ sao cho n_i giải được, ngược lại n_i giải được $\forall i = 1...k$.

- Bài toán n được gọi là không giải được nếu:

+ hoặc n là đỉnh lá và n không phải là đỉnh kết thúc.

+ hoặc T(n)={n_1, n_2,..., n_k}và nếu n là đỉnh HOẶC $\Rightarrow \exists i \in (1...k)$ sao cho n_j không giải được, ngược lại n_i không giải được $\forall i = 1...k$.

- Để tìm lời giải của bài toán khi được phân rã về đồ thị VÀ/HOẶC không phải tìm đường đi mà phải đi tìm đồ thị con gọi là đồ thị con lời giải (hay cây lời giải)

Cây lời giải là đồ thị con G' của G thoả:

- Đỉnh gốc (xuất phát) $n_0 \in V'$

- $\forall n \in V'$, n giải được.

- Ta có sự tương quan:

Phân rã bài toán	Đồ thị VÀ/HOẶC
Bài toán	Đỉnh
Chuyển bài toán thành các bài toán con	Cung
Bài toán sơ cấp	Đỉnh cuối
Các bài toán con phụ	Đỉnh VÀ
Các bài toán con độc lập	Đỉnh HOẶC
Giải bài toán	Tìm đồ thị con lời giải bài toán

3. Các phương pháp tìm kiếm lời giải trên đồ thị và/hoặc.

Sau khi lựa chọn mô tả bài toán và các toán tử quy bài toán về bài toán con, ta có thể xây dựng đồ thị Và/hoặc để giải quyết bài toán ban đầu hoặc chứng tỏ tính không giải được của nó. Cũng như đồ thị trong không gian trạng thái, đồ thị và/hoặc có thể cho dưới dạng tường minh hoặc không tường minh trên cơ sở toán tử xây dựng.

Các phương pháp tìm kiếm trên đồ thị và/hoặc khác nhau chủ yếu ở phương pháp lựa chọn và sắp xếp đỉnh trước khi tháo chúng. Tương tự như trong không gian trạng thái, ta cung có các phương pháp sau:

- Tìm kiếm theo chiều rộng.
- Tìm kiếm theo chiều sâu.
- Tìm kiếm cây lời giải có giá nhỏ nhất.

Các quá trình này khác hẳn với các quá trình lựa chọn trong không gian trạng thái. Sự khác biệt chủ yếu là do việc kiểm tra tính kết thúc của quá trình tìm kiếm và các phương pháp sắp xếp và lựa chọn đỉnh để xét phức tạp hơn nhiều.. Thay cho việc tìm kiếm đỉnh thoả mãn điều kện đích, chúng ta phải tiến hành tìm kiếm đồ thị lời giải. Do đó, ở những thời điểm nhất định, ta phải kiểm tra xem đỉnh đầu có giải được hay không, nếu đỉnh đầu giải được thì kết thúc công việc, trong trường hợp ngược lại thì tiếp tục xét các nút. Nếu đỉnh đang xét không phải là đỉnh kết thúc và nó là đỉnh lá, tức là đỉnh không giải được. Lúc này phải kiểm tra đỉnh đàu có phải không giải được hay không, nếu đúng thì dừng, ngược lại, tiếp tục tìm kiếm.

Trước khi tìm kiếm lời giải trong đồ thị VÀ/HOẶC, chúng ta xây dựng các hàm kiểm tra một đỉnh n nào đó tại thời điểm đang xét có giải được hay không giải được không?

```
Function  giaiduoc(n):boolean;
Begin
        If  <n so cap>  then
                giaiduoc:=true
        else
                if  T(n)<>null then
```

if T(n)$\subset$V$_A$ then

$$giaiduoc = \underset{m \in T(n)}{and}(giaiduoc(m))$$

else

$$giaiduoc = \underset{m \in T(n)}{or}(giaiduoc(m))$$

else

giaiduoc:=false;

End;

Function khonggd(n):boolean;

Begin

If T(n)<>null then

if T(n)$\subset$V$_A$ then

$$khonggd = \underset{m \in T(n)}{or}(khonggd(m))$$

else

$$khonggd = \underset{m \in T(n)}{and}(khonggd(m))$$

else

if <T(n) khong so cap> then

khonggd:=true

else

khonggd:=false;

End;

3.1. Phương pháp tìm kiếm chiều rộng:

Procedure TKR;

Begin

Push(n$_0$, MO);

While MO<>null do