

MỘT PHƯƠNG PHÁP XỬ LÝ CÁC BIỂU THỨC SỐ HỌC TRONG CÁC HỆ TRỢ GIÚP TÍNH TOÁN KHÔNG CẦN LẬP TRÌNH

HOÀNG KIỂM - NGUYỄN MẠNH QUYỀN

Viện Tin học

I - MỞ ĐẦU

Trong thời gian gần đây, một loạt các sản phẩm trợ giúp tính toán không cần lập trình đã xuất hiện trên thị trường thế giới như MathCAD, EUREKA, Mathematica... với mục đích giúp các nhà thiết kế, kỹ sư, học sinh... trong việc tiết kiệm thời gian, đỡ tốn công sức tính toán những công thức số học, đại số thông thường.

Để tăng tốc độ xử lý biểu thức số học, nhất là khi tính toán các biểu thức dạng vòng lặp như tích phân, sigma, tích hay khảo sát hàm số theo giá trị thì việc xem xét thuật toán xử lý biểu thức số học là một việc làm hết sức cần thiết.

II - PHƯƠNG PHÁP TIẾP CẬN

Có nhiều cách tiếp cận khác nhau để tính giá trị cho một biểu thức. Chẳng hạn như là việc xây dựng một chương trình dịch nhỏ để phân tích biểu thức dưới dạng ký hiệu một lần và dựa trên kết quả phân tích đó để tính các lần tiếp sau. Hoặc là cách tính giá trị của biểu thức bằng cách gán giá trị vào nơi đã tính xong cho đến khi không còn hàm hoặc toán tử nữa.

Trong bài viết này sẽ đề cập đến vấn đề phân tích biểu thức một lần duy nhất và có thể tính nhiều lần. Kết quả của cách tiếp cận này đã được thực hiện trong sản phẩm TINCO (Tutor for Intelligent Computing).

1. Xây dựng cây cú pháp

Với mỗi biểu thức có thể biểu diễn chúng dưới dạng một cây cú pháp nhị phân.

Ví dụ 1

Hình 2 là cây biểu diễn của biểu thức

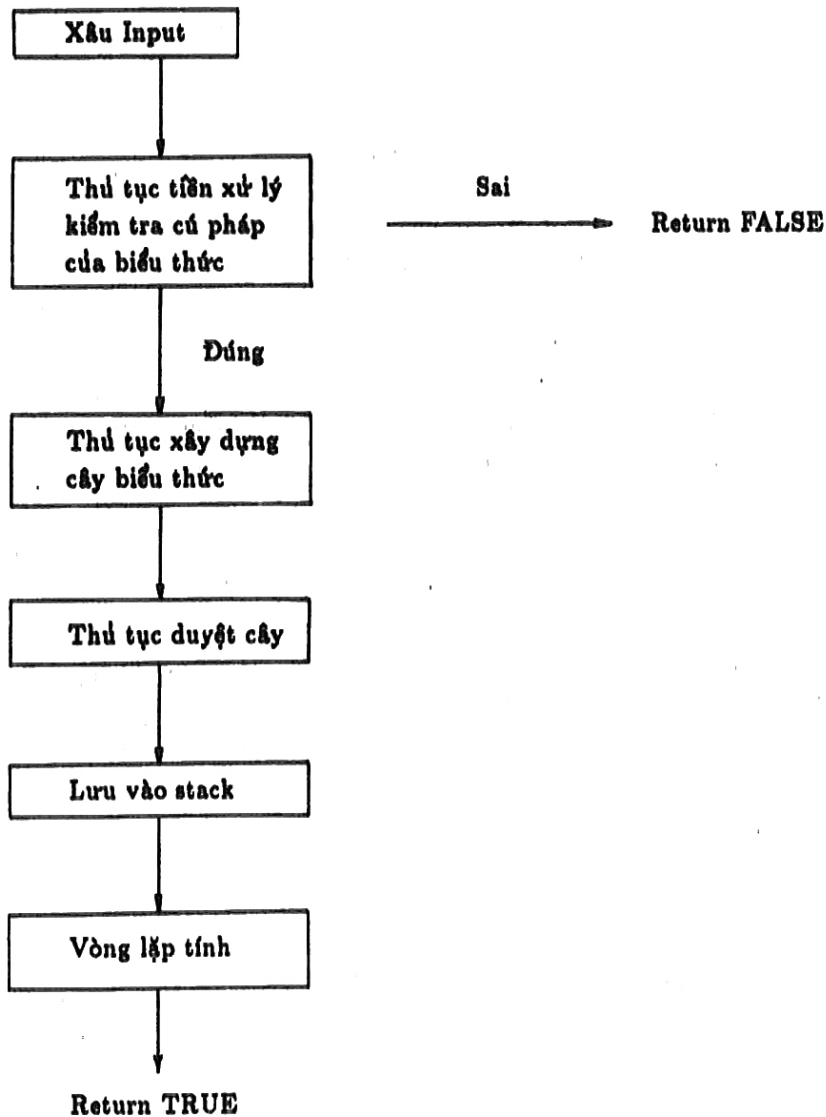
$$(a - d * e) * \sin(x + a \wedge d)$$

Trong cây vì cấu trúc phân cấp của nó nên các dấu ngoặc đóng và mở không được thể hiện như là các lá của cây. Các hàm chuẩn hoặc là một số toán tử cho phép tác động một ngôi như là toán tử "-" cũng có được thể hiện như là một đỉnh với một lá của nó là NULL.

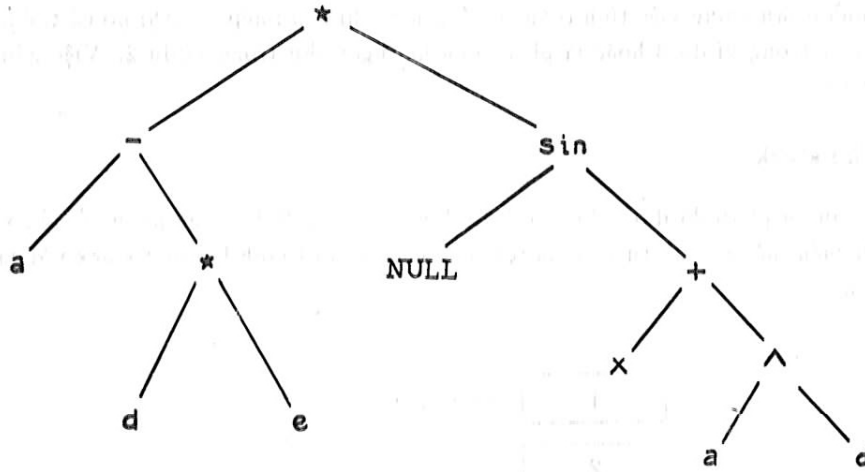
Ví dụ 2:

Hình 3 biểu diễn biểu thức

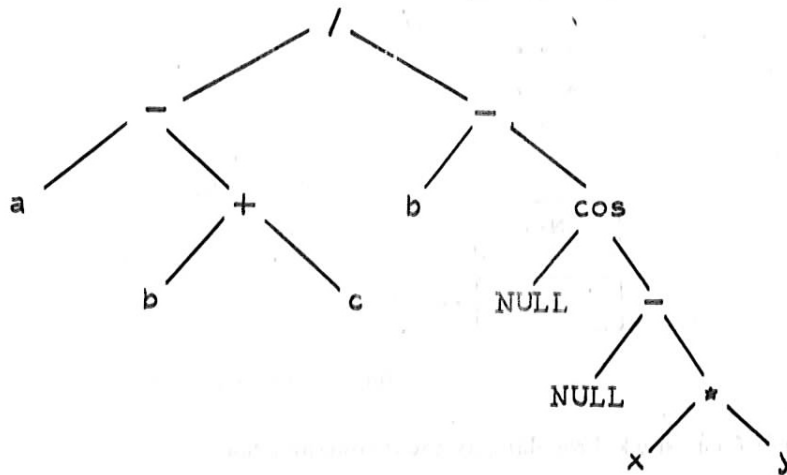
$$a - (b + c) / (b - \cos(-x * y))$$



Hình 1: Lược đồ quá trình phân tích cây biểu thức



Hình 2



Hình 3

2. Duyệt cây

Sau khi xây dựng cây cú pháp cho biểu thức, quá trình duyệt cây theo thứ tự trái qua phải được thực hiện và được lưu theo dạng ký pháp Balan sau khi duyệt.

Chẳng hạn, các cây cú pháp ở hình 2 và 3 được duyệt với kết quả như dưới đây:

$$a d e * - x a d \wedge + \sin *$$

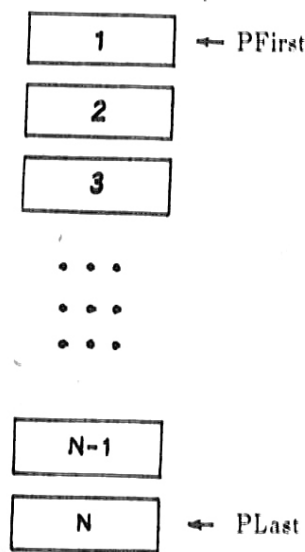
$$a b c + - b x y * - \cos - /$$

Dạng ký pháp Balan là dạng biểu diễn duy nhất cho mỗi biểu thức bất kỳ.

Với một số đỉnh chỉ có một lá nhất thiết phải gán cho chúng một thuộc tính đặc biệt để tránh sự nhầm lẫn trong việc tính toán. Chẳng hạn khi gặp phép “-” thì nó có thể là phép toán một ngôi như trong ví dụ 3 hoặc là phép toán hai ngôi như trong ví dụ 2. Việc gán thuộc tính này là bắt buộc.

3. Lưu vào stack

Với xâu cú pháp đã được đưa ra dưới dạng ký pháp Balan, các phần tử của xâu ký pháp này là một biến, hằng, toán tử hoặc là tên hàm chuẩn. Ta thành lập một stack với cấu trúc móc nối như sau:



Hình 4

Mỗi một ô của stack được đăng ký các trường như sau:

- string Name
- byte Attribute
- pointer Next
- pointer Last

Name: Là xâu ký tự lưu tên các phần tử được tách ra từ xâu ký pháp sau khi phân tích.

Attribute: Thuộc tính của Name, có thể là biến, hằng, toán tử hoặc là tên hàm chuẩn.

Các pointer Next, Last: Là pointer của ô hiện thời trỏ đến các ô sau hoặc trước trong stack tương ứng.

4. Thực hiện tính toán trên stack

Với cấu trúc được thành lập như trên, quá trình tính toán sẽ được thực hiện theo đoạn mô

XỬ LÝ CÁC BIỂU THỨC SỐ HỌC TRONG CÁC HỆ TRỢ GIÚP TÍNH TOÁN

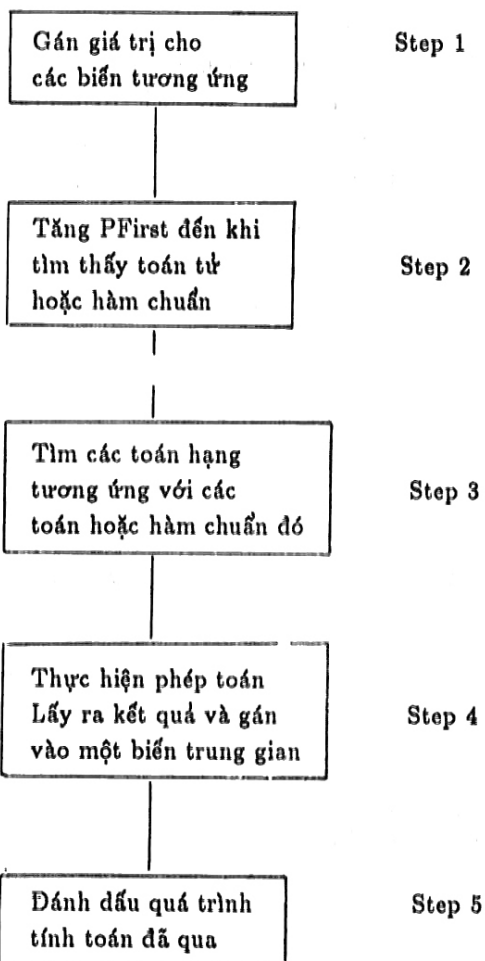
tả bằng thủ tục sau đây:

Gọi PFirst là pointer trở tới ô đầu tiên trong stack,

PEnd là pointer trở ô cuối cùng trong stack.

while (PFirst <> PEnd)

begin



end

Hình 5

Giá trị cuối cùng trong ô nhớ trung gian chính là giá trị của biểu thức đã cho.

5. Tính toán với các biểu thức hỗn hợp gồm có sigma, tích

Giá trị với một biểu thức đã cho có dạng:

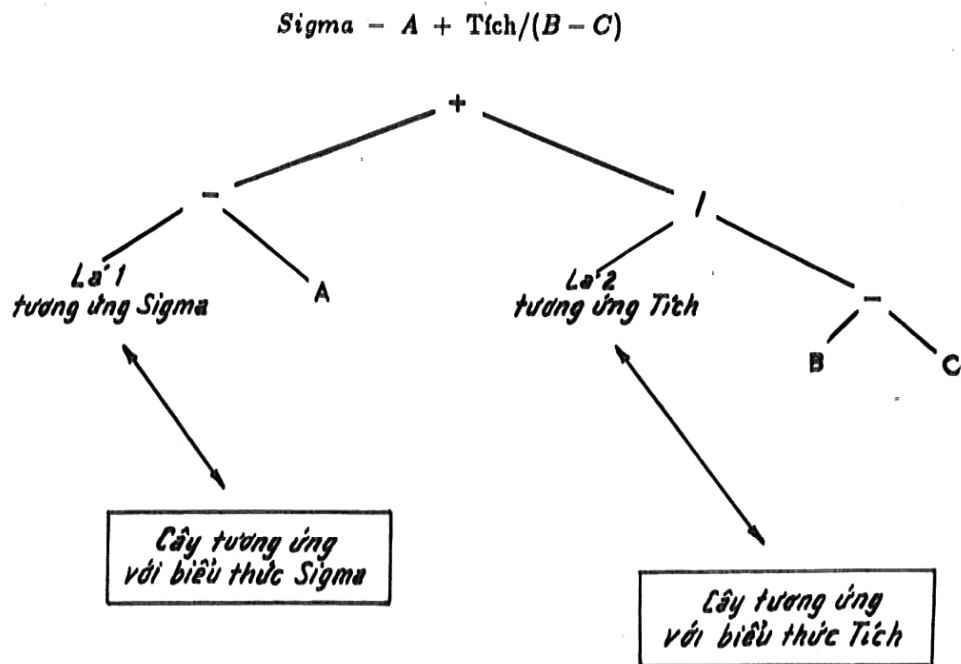
$$F_1(x_1, x_2, \dots) + \text{Sigma}(i = m..n, S(i)) * F_2(y_1, y_2, \dots) - \text{Tích}(j = 1..k T(k))$$

lúc đó coi Sigma như là một số hạng trong xâu biểu thức, ta vẫn có một cây cú pháp tương ứng với nó và Sigma hay Tích như là những lá hình thức trong cây.

Các bước thực hiện sẽ theo thứ tự sau:

- Thành lập cây biểu thức chính.
- Thành lập các cây Sigma và tích nếu chúng tồn tại.
- Thiết lập các pointer từ các lá là Sigma hay Tích trở đến các cây tương ứng.
- Quá trình tính toán được thực hiện từ các cây Sigma và Tích ngược trở lên, các giá trị được gán vào các lá tương ứng với chúng.

Ví dụ: Hình 6 là cây biểu diễn từ biểu thức



Hình 6.

Để thực hiện tính toán trên các cây Sigma hay Tích thì quá trình thực hiện sẽ là như sau:

- Tách các biểu thức nằm trong Sigma hay Tích.
- Tách chỉ số, cận trên và cận dưới.
- Thành lập cây cú pháp từ biểu thức vừa tách.
- Phân tích và lưu vào stack.
- Thực hiện vòng lặp tính.

Giả sử: i là chỉ số; m, n là các cận dưới và trên.

Thủ tục tính toán được mô tả như sau cho trường hợp tính Tích:

```
i = m
Tích = 1;
while (i ≤ n)
  begin
    Tích = Tích * kết quả Tính toán trên stack;
    i++;
  end
```

Giá trị tính toán được từ các cây Sigma hay Tích sẽ được lưu vào các lá tương ứng với chúng trong cây biểu thức chính.

III - KẾT LUẬN

Với thủ tục được xây dựng như trên, việc tính toán đã được tăng tốc độ xử lý lên nhiều lần khi mà độ phức tạp của công thức tăng lên do số ngoặc và số các toán tử.

Việc phân tích biểu thức dưới dạng ký hiệu một lần và tính nhiều lần sau dựa trên thứ tự tuần tự của stack đã được thành lập còn đảm bảo cả về độ chính xác của kết quả phép toán như là máy tính cho phép.

Nhận ngày 15 - 12 - 1991

TÀI LIỆU THAM KHẢO

1. Bạch Hưng Khang - Hoàng Kiếm, Trí tuệ nhân tạo: Các phương pháp và ứng dụng. NXB Khoa học Kỹ thuật Hà Nội, 1989.
2. Nilson N. J., Problem solving methods in AI. Newyork, 1986.
3. 3C-Soft - Tutor for Intelligent Computing. Hanoi, 1991.

ABSTRACT

One method for numeric string processing in the Tutor of intelligent and non-programable computing

This paper deals with the programming techniques for numeric string processing in the mathematical tutor systems. The keys of the method is a technique to sparse basic parts of the numeric string. Some applications in implementation of this method are given.