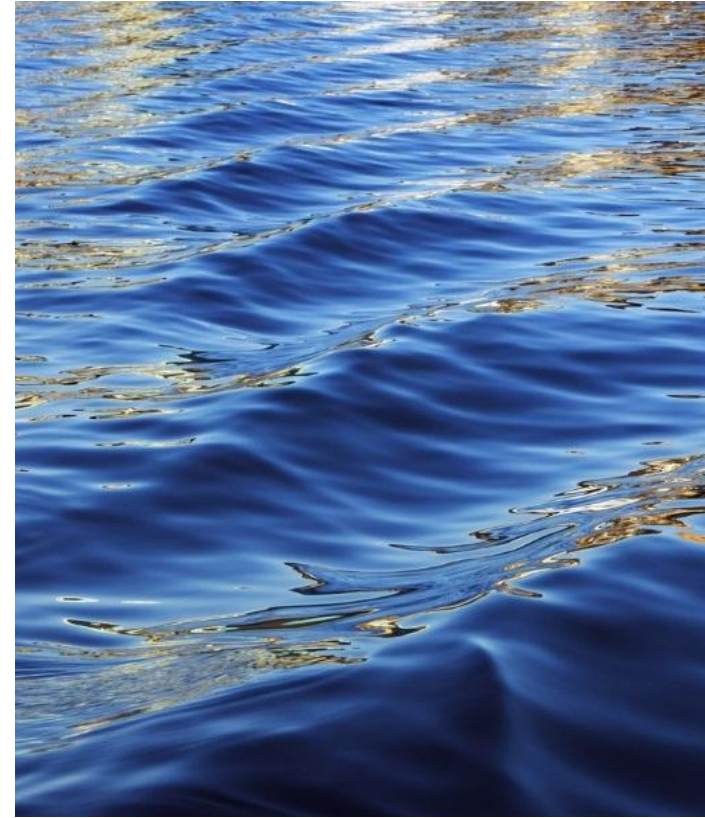




I. Genetic Programming






I. Giới thiệu về Genetic Programming



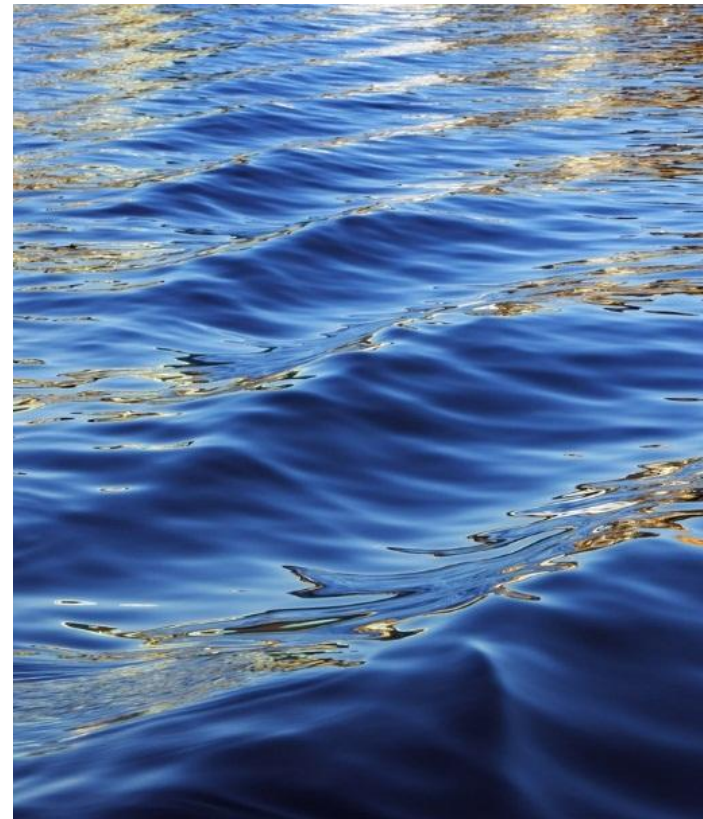
I. Giới thiệu về Genetic Programming

- Genetic Programming(GP) được giới thiệu bởi Mr. John Koza là 1 loại thuộc Evolutionary Algorithm(EA), một nhánh nhỏ của machine learning. EAs thường được sử dụng để tìm ra giải pháp cho những bài toán mà con người chưa thể tìm ra được cách giải một cách trực tiếp.
- Genetic Programming là 1 phương pháp có hệ thống giúp máy tính tự động giải quyết vấn đề và chuyển đổi lặp đi lặp lại 1 tập hợp tất cả các lời giải có thể của chương trình máy tính sang 1 thế hệ các chương trình mới bằng cách tương tự như các hoạt động di truyền xảy ra 1 cách tự nhiên. Các hoạt động di truyền bao gồm: crossover(lai ghép), mutation(đột biến), reproduction(sinh mới), gene duplication(lặp gen) and gene deletion(xóa gen).
- Nói chung Genetic Programming là sự tiến hóa của các chương trình máy tính.



II. Quá trình hoạt động của Genetic Programming

Subtitle

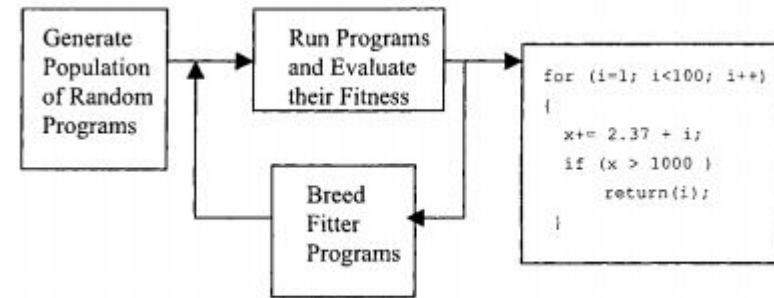


II. Quá trình hoạt động của Genetic Programming

- Genetic Programming được bắt đầu bởi một tập các chương trình khởi tạo bao gồm có functions(các hàm) và terminals(các điểm cuối) có thể được chọn tay từ trước hoặc sinh ra một cách ngẫu nhiên. Các hàm có thể là các toán tử số học tiêu chuẩn, các toán tử lập trình, các hàm số toán học hoặc các hàm số logic. Các chương trình cạnh tranh với nhau qua dữ liệu đầu vào và đầu ra kì vọng. Mỗi chương trình máy tính trong tập hợp tất cả các giải pháp được đánh giá thông qua khả năng hoạt động của nó như thế nào trong các môi trường vấn đề cụ thể. Sự đánh giá này gọi là fitness measure(thước đo sự thích nghi) Các chương trình có hiệu suất hoạt động tốt nhất sẽ được chọn, sự đột biến và sản sinh sẽ được thực hiện trên chúng để sinh ra thế hệ tiếp theo. Thế hệ tiếp theo sẽ cạnh tranh với các chương trình khác, quá trình sẽ tiếp tục cho đến khi chương trình hoàn hảo được phát triển.

- Có thể giải thích 1 cách ngắn gọn như sau:

- Tạo 1 tập các giải pháp khởi tạo.
- Đánh giá: gán 1 hàm tính toán sự tương thích vào mỗi chương trình trong tập các giải pháp.
- Lựa chọn ra các giải pháp tương thích hơn.
- Tạo ra các biến thể mới(bằng cách lai ghép, đột biến, ..)
- Lặp lại các bước trên.
- Kết thúc.



Vòng lặp chính của Genetic Programming



III. Biểu diễn các giải pháp

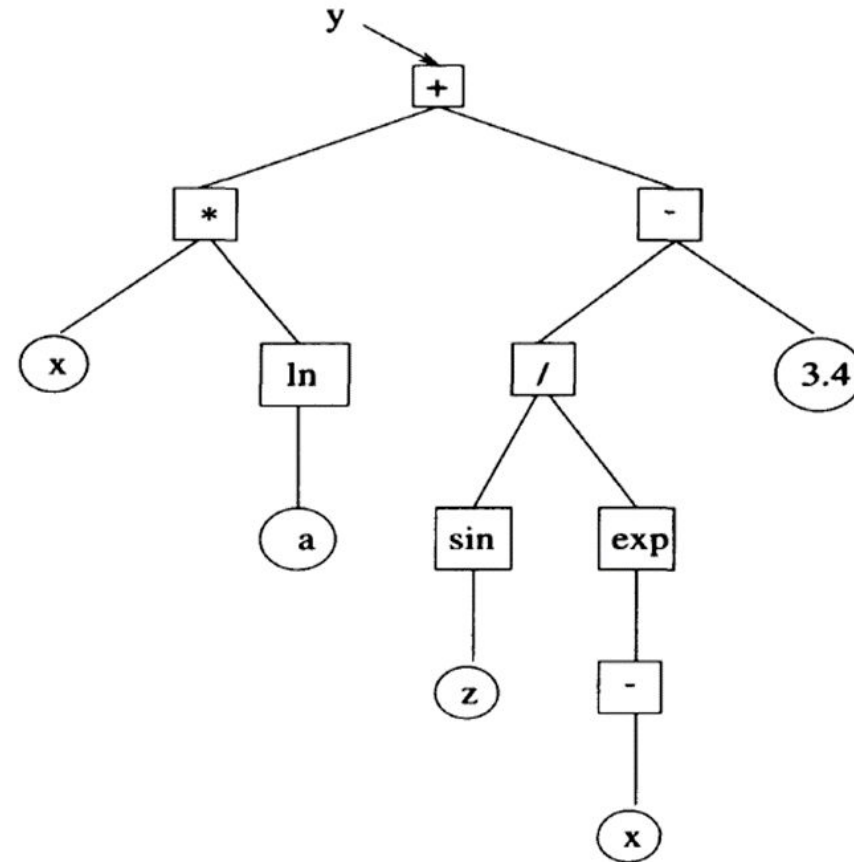



III. Biểu diễn các giải pháp

- Mỗi giải pháp trong GP biểu diễn một chương trình máy tính hay một hàm ẩn cần tìm.
- Để đảm bảo đúng cú pháp cho giải pháp, ta cần xác định 2 tập:
 - Tập kết thúc (terminal set) chứa các biến, hằng số, modun cơ bản có thể của chương trình
 - Tập hàm (function set) chứa tất cả các hàm toán học cơ bản có thể dùng: +, -, *, /, exp, log, and, or, xor, ... hoặc các cấu trúc quyết định if-then-else, ...
- Mỗi giải pháp được thể hiện dưới dạng một cấu trúc cây:
 - Các nút lá của cây: Chọn từ tập kết thúc.
 - Các nút trong của cây: Chọn từ tập hàm.
- Mỗi giải pháp có thể được khởi tạo như sau:
 - Nút gốc: được chọn ngẫu nhiên trong tập hàm
 - Nút không phải gốc: chọn ngẫu nhiên trong tập hàm hoặc tập kết thúc.
 - Chọn trong tập kết thúc -> nút trở thành nút lá
 - Chọn trong tập hàm -> nút là nút trong
 - Số lượng nhánh tại mỗi nút phụ thuộc vào số lượng các tham số của hàm tại nút đó.

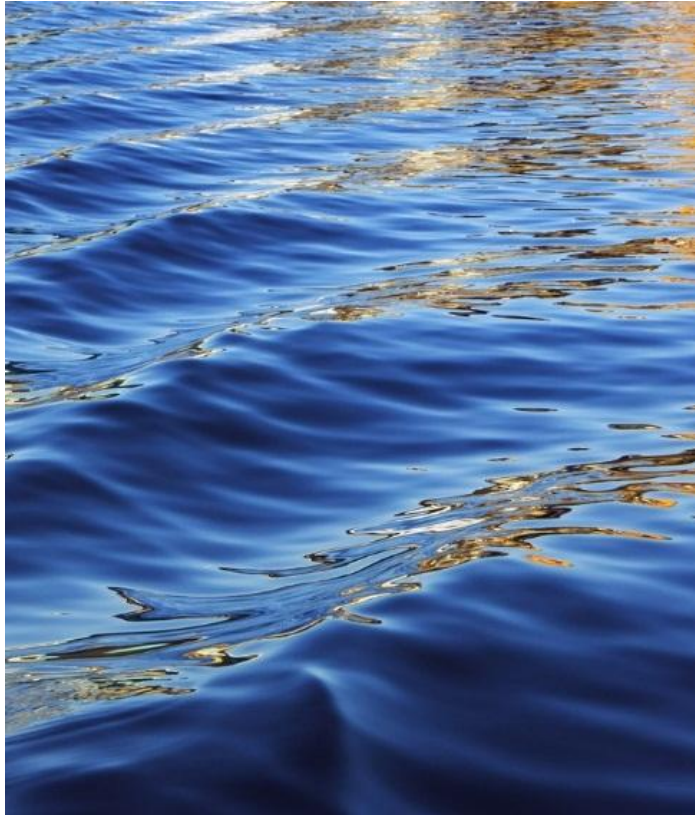
Biểu diễn giải pháp – Ví dụ

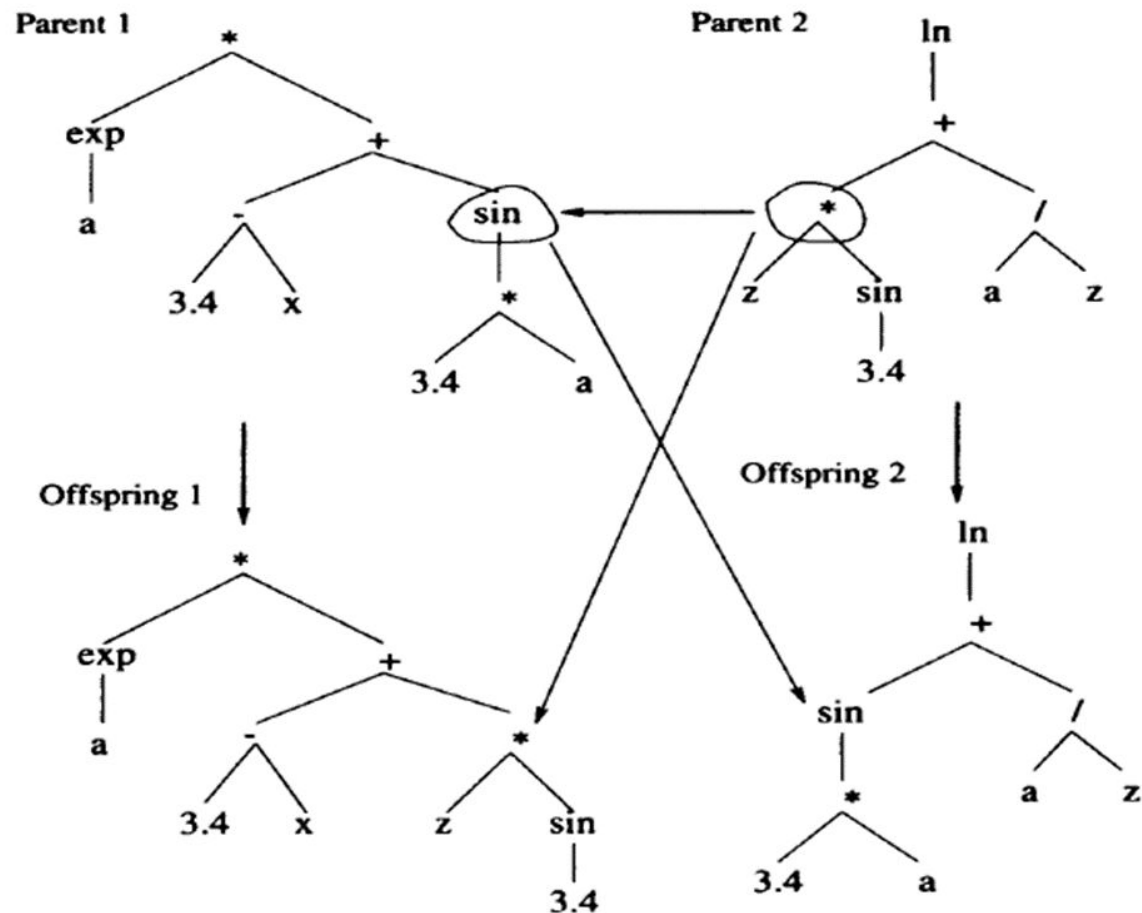
- $y := x * \ln(a) + \sin(z) / \exp(-x) - 3.4$
- Tập kết thúc: $x, a, z, 3.4$
- Tập hàm: $*, +, -, /, \ln, \sin, \exp$





IV. Các toán tử tiến hóa(Genetic operators)



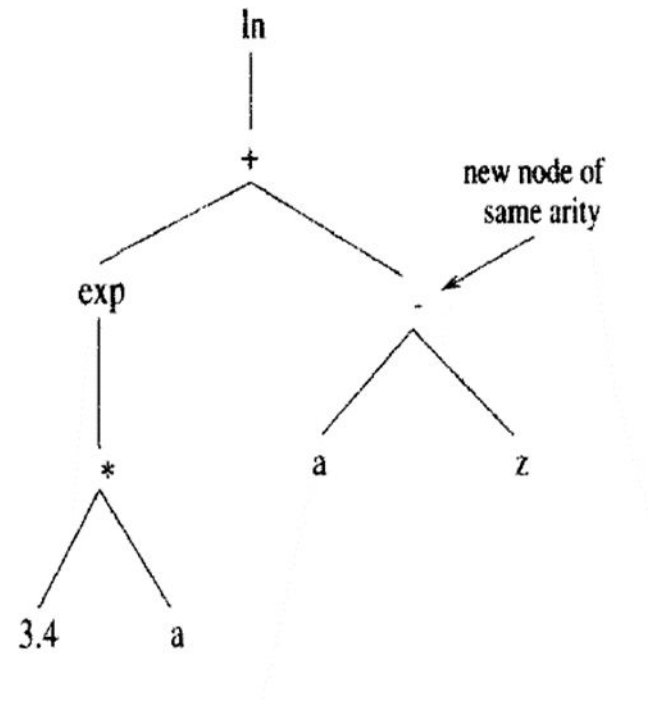
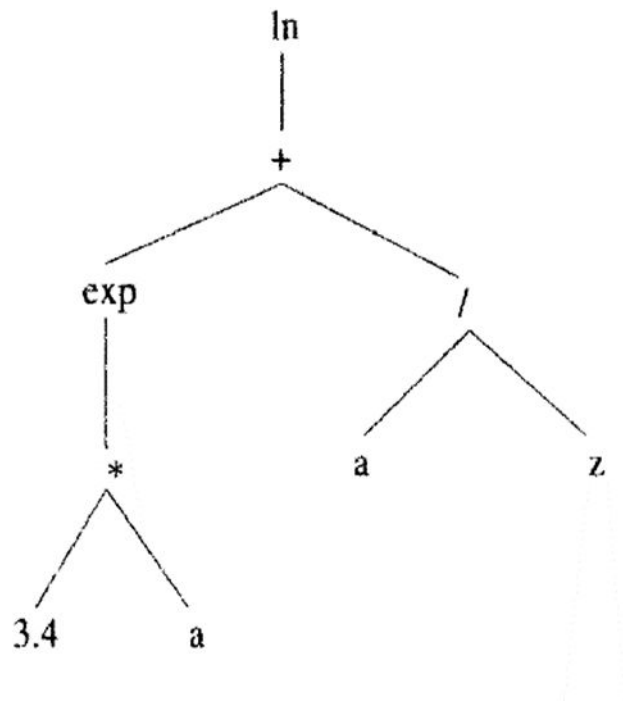


Ví dụ

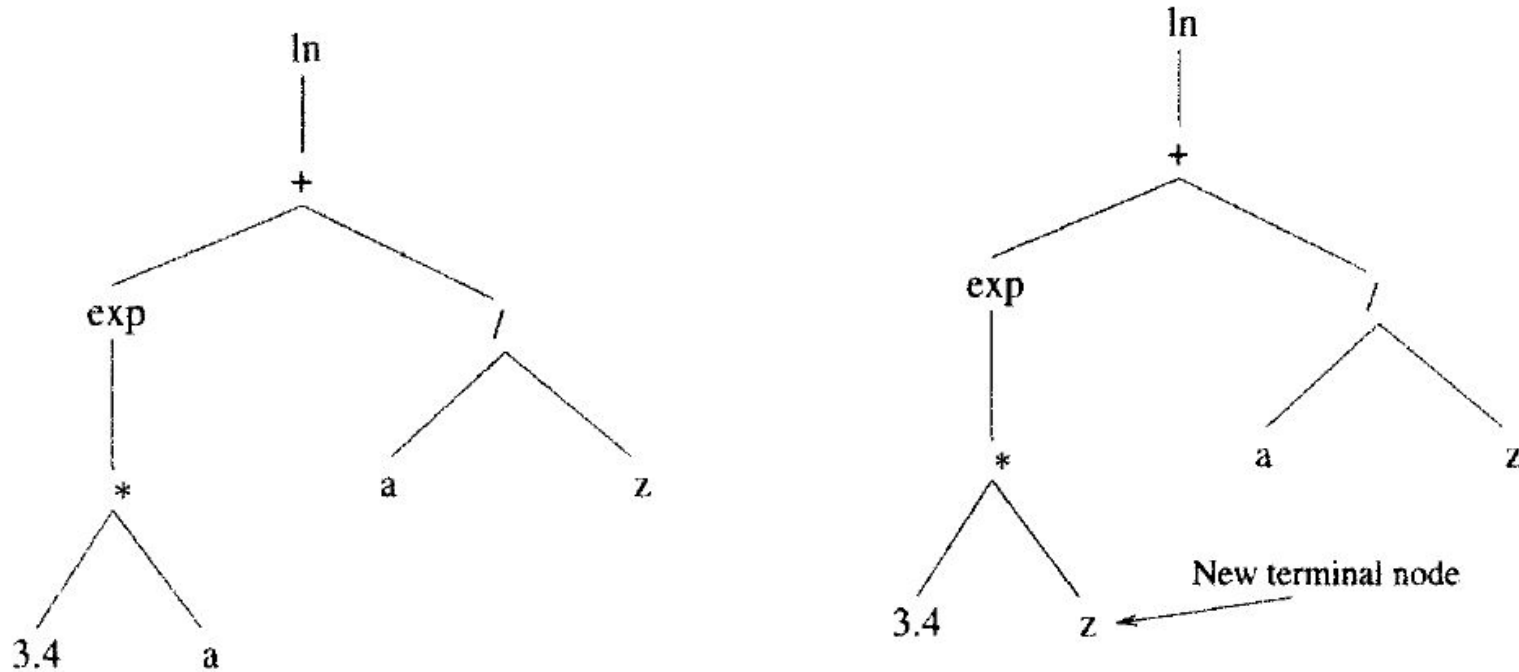
Toán tử lai ghép

Chọn ngẫu nhiên 1 cây con trong 1 cây cha mẹ và trao đổi cho 1 cây con trong 1 cây cha mẹ khác

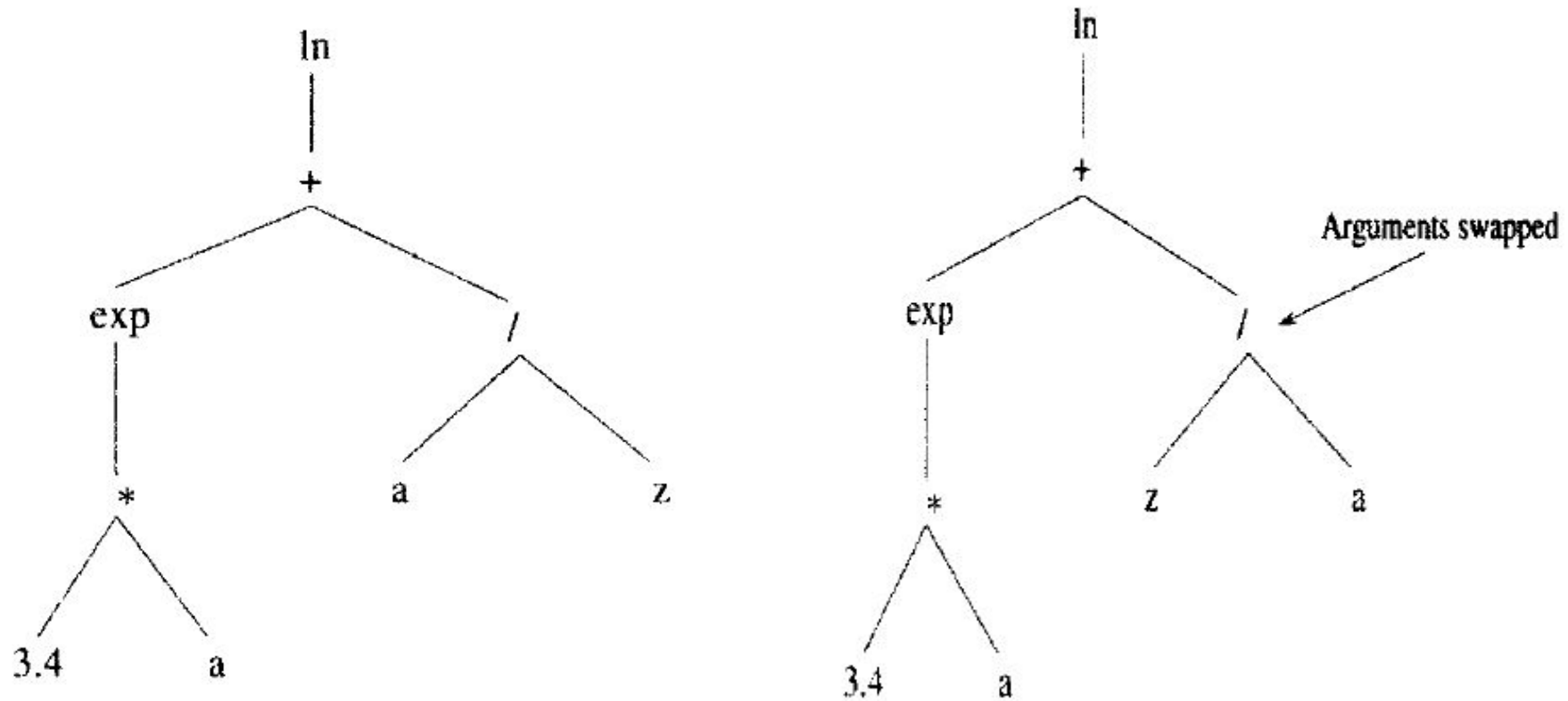
Đột biến-Đột biến nút trong: Thay thế hàm tại nút đó bằng 1 hàm khác trong tập hàm.



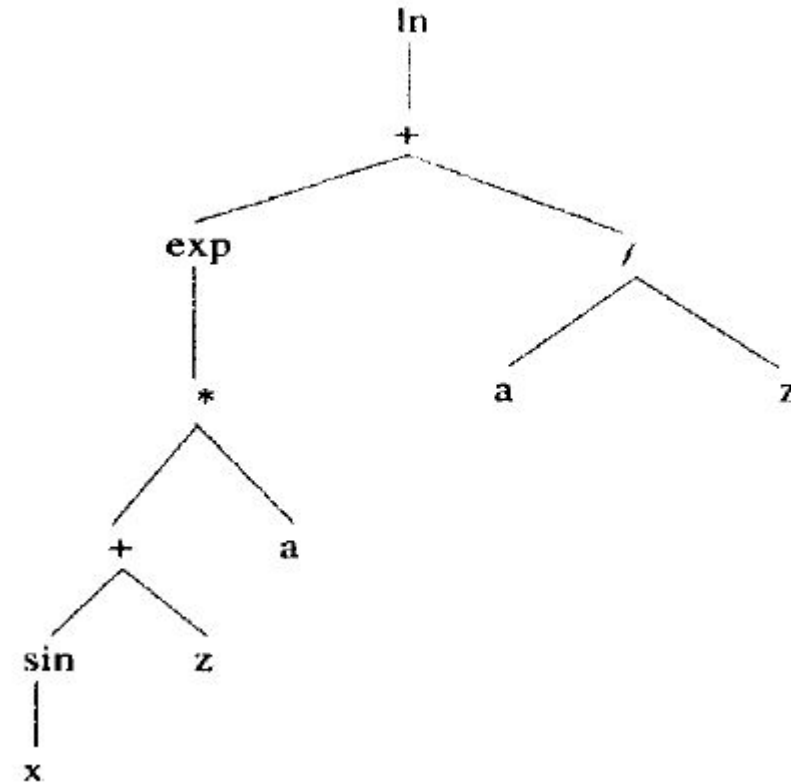
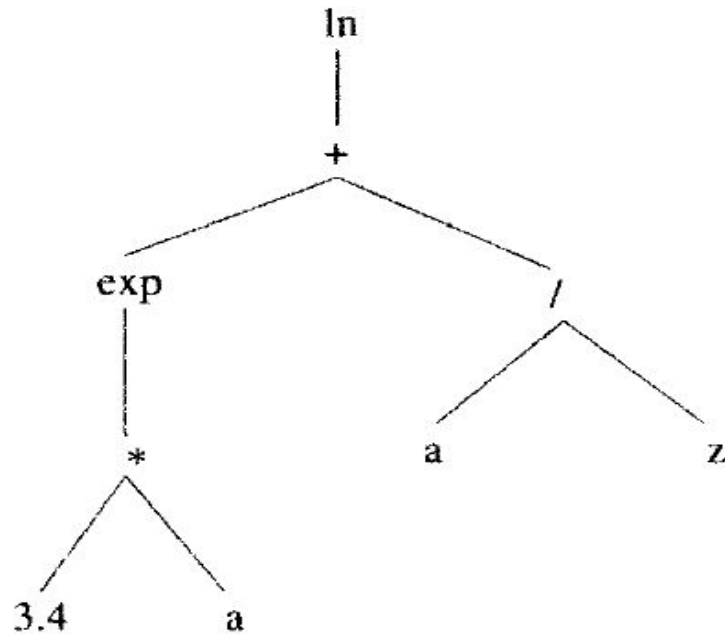
Đột biến nút kết thúc: Thay thế biến, hằng tại nút đó bằng một biến, hằng khác trong tập kết thúc.



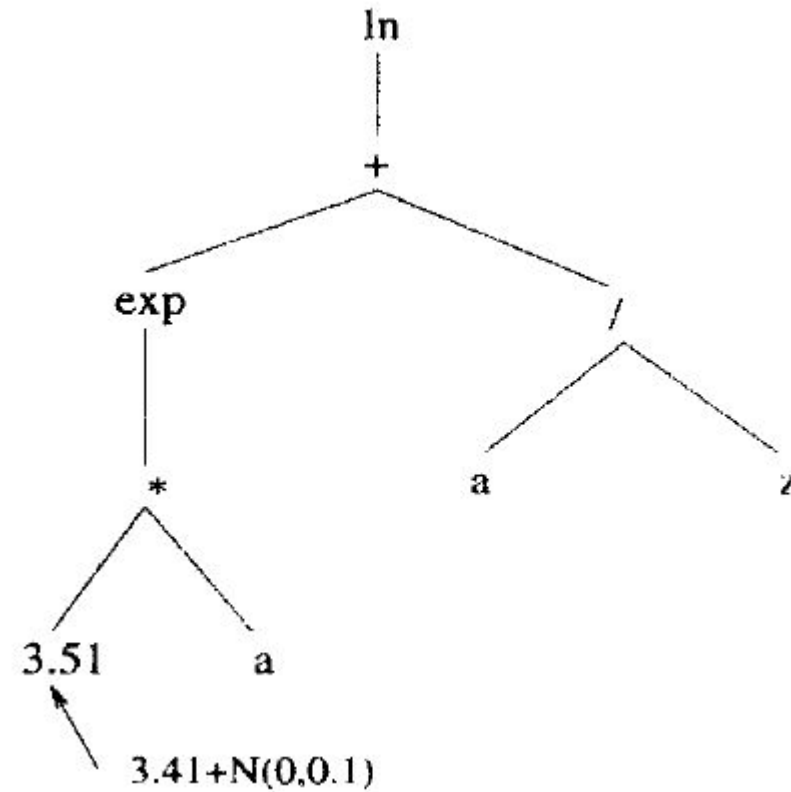
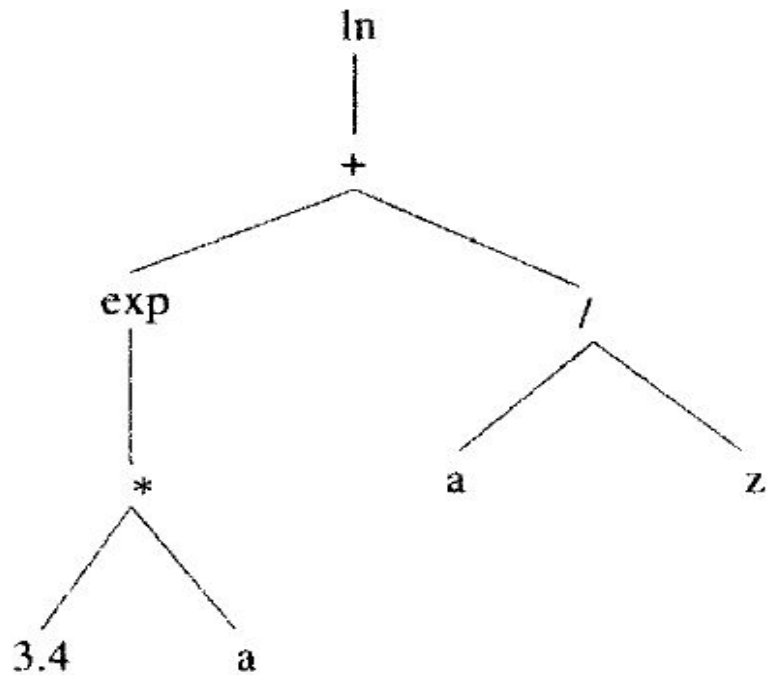
Đột biến-Đột biến đảo: Chọn ngẫu nhiên 1 nút trong và đảo vị trí các nút con của nó.



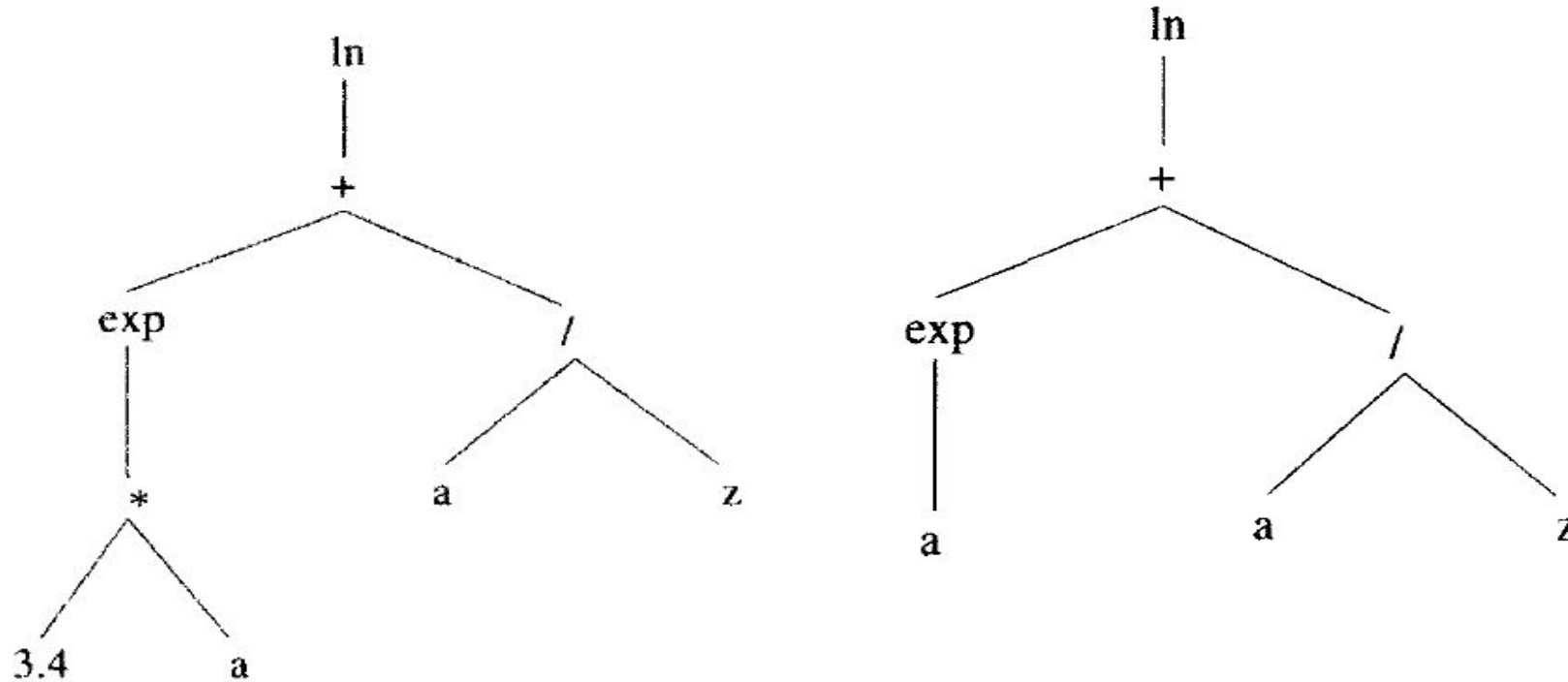
Đột biến-Đột biến phát triển cây: Chọn ngẫu nhiên 1 nút và thay thế toàn bộ cây con của nút đó bằng một cây con ngẫu nhiên khác.



Đột biến-Đột biến Gauss: Chọn ngẫu nhiên 1 nút lá chứa hằng số và thêm giá trị nhiễu Gauss vào nút đó.



Đột biến-Đột biến cắt tỉa cây: Chọn ngẫu nhiên 1 nút và thay thế nút đó bởi 1 biến hay hằng số ngẫu nhiên trong tập kết thúc.





V. Đánh giá độ thích nghi



Đánh giá độ thích nghi

- Các giải pháp được đánh giá trên cùng tập mẫu dữ liệu và giá trị hiệu suất trung bình thu được trên mẫu đó được sử dụng là giá trị độ thích nghi.
- Giả sử có 1 tập mẫu X và mỗi mẫu chứa 3 giá trị ban đầu vào (a, x, z) và giá trị đích là y . Độ thích nghi được tính như sau:
 - Tính giá trị đầu ra y' thu được của chương trình mà giải pháp biểu diễn với mỗi đầu vào (a, x, z)
 - Tính giá trị lỗi của y' so với y .
 - Độ sai số trung bình trên toàn bộ tập dữ liệu được xem như tiêu chí để đánh giá độ thích nghi.



Cảm ơn mọi người
đã lắng nghe.

