

Cấu trúc dữ liệu và giải thuật

TS. Nguyễn Văn Đoàn

Chương 2 – Biểu thức toán học

Ứng dụng Stack tính toán biểu thức toàn học

Thuật toán Ba Lan ngược

- Thuật toán Ba Lan ngược
(Reverse Polish Notation – RPN)
 - ▣ Định nghĩa RPN:
 - Biểu thức toán học trong đó các toán tử được viết sau toán hạng và không dùng dấu ngoặc
 - ▣ Phát minh bởi Jan Lukasiewics một nhà khoa học Ba Lan vào những năm 1950

Thuật toán Ba Lan ngược

RPN

Infix : toán tử viết **giữa** toán hạng
Postfix (RPN): toán tử viết **sau** toán hạng
Prefix : toán tử viết **trước** toán hạng

Examples:

INFIX

RPN (POSTFIX)

PREFIX

A + B

A B +

+ A B

A * B + C

A B * C +

+ * A B C

A * (B + C)

A B C + *

* A + B C

A - (B - (C - D))

A B C D - - -

- A - B - C D

A - B - C - D

A B - C - D -

- - - A B C D

Lượng giá biểu thức RPN

Kỹ thuật gạch dưới:

1. Duyệt từ trái sang phải của biểu thức cho đến khi gặp toán tử.
2. Gạch dưới 2 toán hạng ngay trước toán tử và kết hợp chúng bằng toán tử trên
3. Lặp đi lặp lại cho đến hết biểu thức.

Ví dụ $2*((3+4)-(5-6))$

2 3 4 + 5 6 - - *

→ 2 3 4 + 5 6 - - *

→ 2 7 5 6 - - *

→ 2 7 5 6 - - *

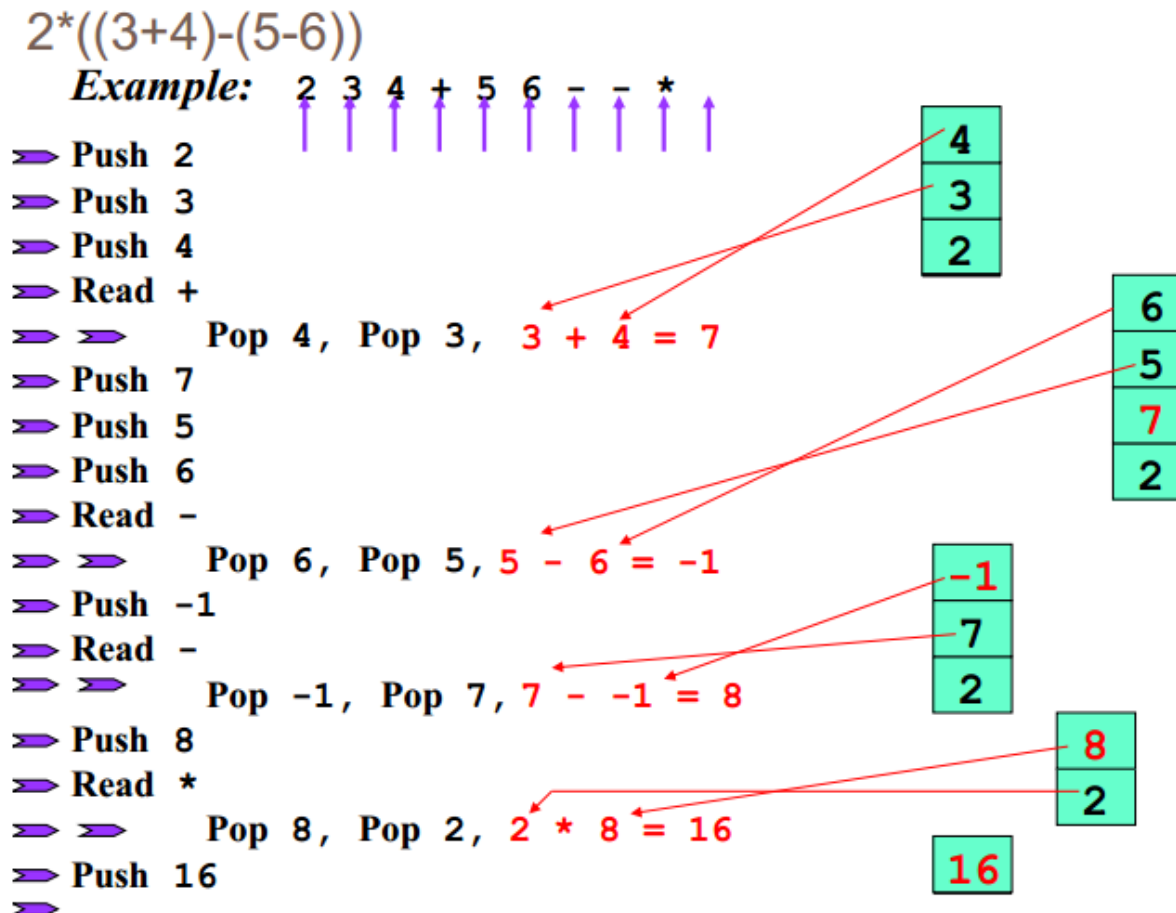
→ 2 7 -1 - *

→ 2 7 -1 - * → 2 8 * → 2 8 * → 16

Thuật toán tính giá trị

1. Khởi tạo Stack rỗng (*chứa hằng hoặc biến*).
2. Lặp cho đến khi kết thúc biểu thức:
 - Đọc 01 phần tử của biểu thức (*hằng, biến, phép toán*).
 - Nếu phần tử là hằng hay biến: đưa vào Stack.
 - Ngược lại:
 - Lấy ra 02 phần tử của Stack.
 - Áp dụng phép toán cho 02 phần tử vừa lấy ra.
 - Đưa kết quả vào Stack.
3. Giá trị của biểu thức chính là phần tử cuối cùng của Stack.

Thuật toán tính giá trị - ví dụ



Độ ưu tiên

□ + , _	1
□ *, /	2
□ ^	3

Chuyển infix thành postfix

1. Khởi tạo Stack rỗng (chứa các phép toán).

2. Lặp cho đến khi kết thúc biểu thức:

 Đọc 01 phần tử của biểu thức

(01 phần tử có thể là hằng, biến, phép toán, “ ”)

hay “(”).

 Nếu phần tử là:

2.1 “(” : đưa vào Stack.

2.2 “)” : lấy các phần tử của Stack ra
cho đến khi gặp “(” trong Stack.

Chuyển infix thành postfix

2.3 Một phép toán: $+$ $-$ $*$ $/$

Nếu **Stack rỗng**: đưa vào Stack.

Nếu Stack khác rỗng và **phép toán có độ ưu tiên cao hơn phần tử ở đầu Stack**: đưa vào Stack.

Nếu Stack khác rỗng và phép toán có **độ ưu tiên thấp hơn hoặc bằng phần tử ở đầu Stack**:

- lấy phần tử từ Stack ra;
- sau đó lặp lại việc so sánh với phần tử ở đầu Stack.

2.4 Hằng hoặc biến: đưa vào kết quả.

3. Lấy hết tất cả các phần tử của Stack ra.

Chuyển infix thành postfix – ví dụ

Example: $(A+B*C) / (D-(E-F))$

➤ Push (
➤ Display A
➤ Push +
➤ Display B
➤ Push *
➤ Display C
➤ Read)
➤ Pop *, Display *,
➤ Pop +, Display +, Pop (
➤ Push /
➤ Push (
➤ Display D
➤ Push -
➤ Push (
➤ Display E
➤ Push -
➤ Display F
➤ Read)
➤ Pop -, Display -, Pop (
➤ Read)
➤ Pop -, Display -, Pop (
➤ Pop /, Display /

Output

A

AB

ABC

ABC*

ABC*+

ABC*+D

ABC*+DE

ABC*+DEF

ABC*+DEF-

ABC*+DEF--

ABC*+DEF--/

