

Data Structure #9

infix to postfix conversion

2020년 1학기

Intro.

- 실습주제 소개
 - Infix to postfix conversion
- 실습수업 문제
 - 연산자 우선순위 설정 + Infix to postfix conversion

CSLAB

수식의 계산

- 수식의 표기방법:

- 전위(prefix), 중위(infix), 후위(postfix)

| 중위 표기법 | 전위 표기법 | 후위 표기법 |
|-----------|---------|---------|
| $2+3*4$ | $+2*34$ | $234*+$ |
| $a*b+5$ | $+5*ab$ | $ab*5+$ |
| $(1+2)+7$ | $+7+12$ | $12+7+$ |

CSLAB

수식의 계산

- 사람의 경우

(중간결과는 temp에 기록한다)

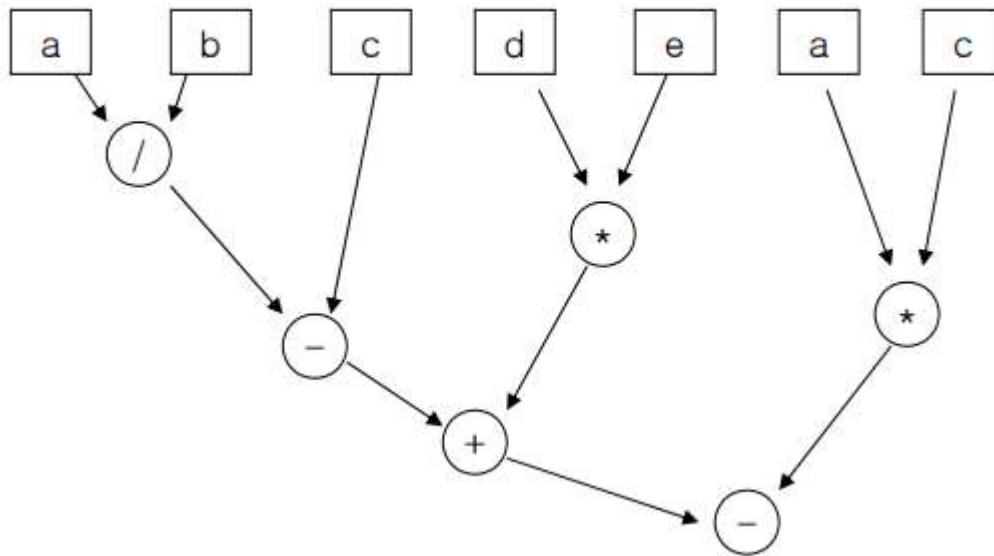
$$\begin{aligned}x &= (a/b) - c + d * e - a * c \\&= (\text{temp1}) - c + d * e - a * c \\&= (\text{temp1}) - c + (\text{temp2}) - a * c \\&= (\text{temp1}) - c + (\text{temp2}) - (\text{temp3}) \\&= (\text{temp4}) + (\text{temp2}) - (\text{temp3}) \\&= (\text{temp5}) - (\text{temp3}) \\&= (\text{temp6})\end{aligned}$$

CSLAB

수식의 계산

- 사람의 경우

$$x = (((a/b) - c) + (d * e)) - (a * c)$$



CSLAB

수식의 계산

- 컴퓨터의 수식 계산

- 중위 표기법으로 표현된 수식을 후위 표기법으로 바꾼다.
- 후위 표기법으로 표현된 수식을 계산한다.

$$- 3 * 5 + 4 \quad -> \quad 3 \ 5 * 4 +$$


CSLAB

중위표기식 -> 후위표기식

- 중위표기와 후위표기

- 중위 표기법과 후위 표기법의 공통점은 피연산자의 순서는 동일
- 연산자들의 순서만 다름(우선순위 순서에 따라)
 - > 연산자만 스택에 저장했다가 출력하면 된다.
- $2+3*4$ -> $234*+$

CSLAB

중위표기식 -> 후위표기식

• 알고리즘

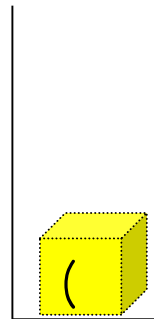
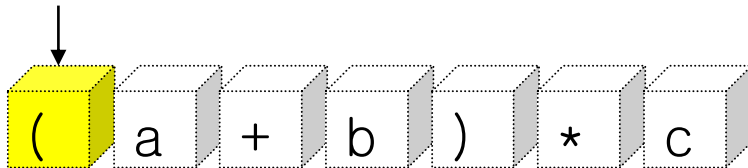
- 피연산자를 만나면 그대로 출력
- 연산자를 만나면 스택에 저장
- 스택에 있는 연산자가 현재 처리중인 연산자 보다 우선순위가 높거나 같으면 먼저 pop하고, push 함. 아니면, 바로 push.
- 왼쪽 괄호는 무조건 스택에 넣고, 스택에 있는 왼쪽 괄호는 우선순위가 가장 낮은 연산자로 취급
- 오른쪽 괄호가 나오면 스택에서 왼쪽 괄호를 포함하여 위에 쌓여있는 모든 연산자를 출력

CSLAB

중위표기식 -> 후위표기식

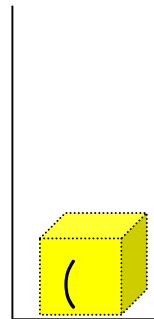
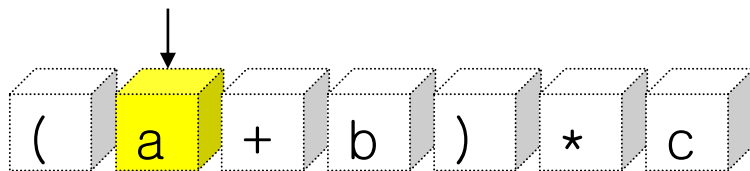
• 후위 표기 수식 변환의 예

- $a + b * c \Rightarrow a b c * +$
- $a * b + c \Rightarrow a b * c +$
- $(a + b) * c \Rightarrow$



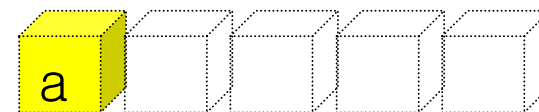
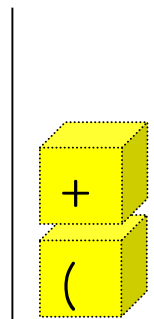
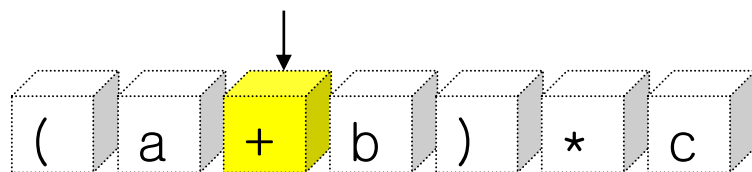
CSLAB

중위표기식 -> 후위표기식



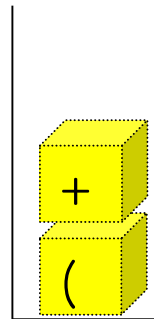
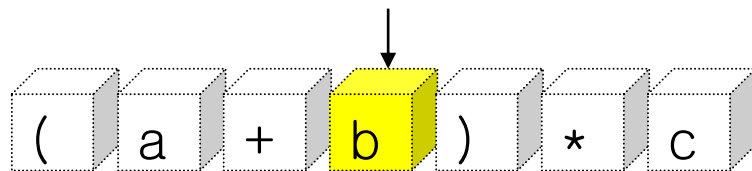
CSLAB

중위표기식 -> 후위표기식



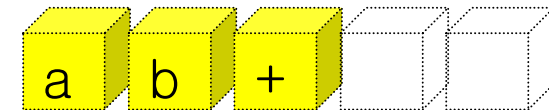
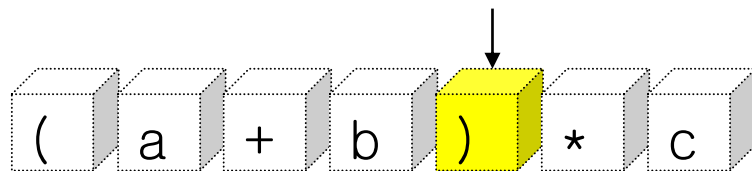
CSLAB

중위표기식 -> 후위표기식



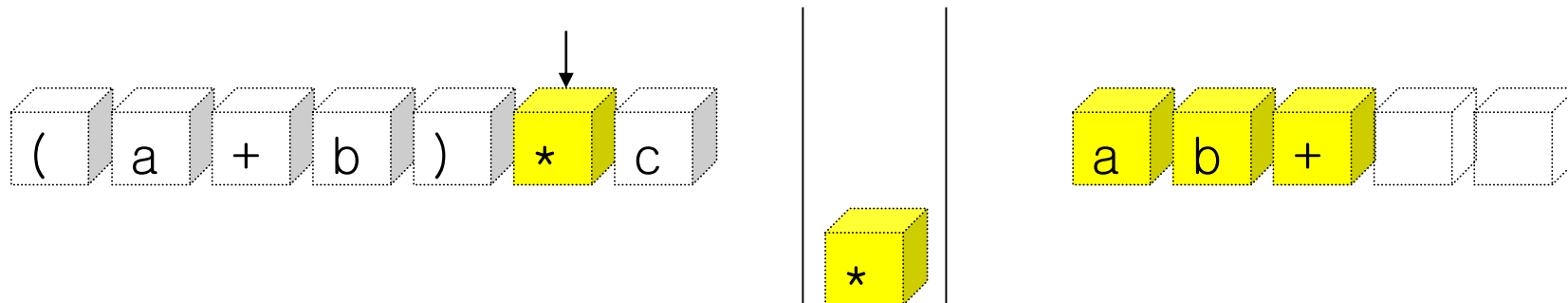
CSLAB

중위표기식 -> 후위표기식



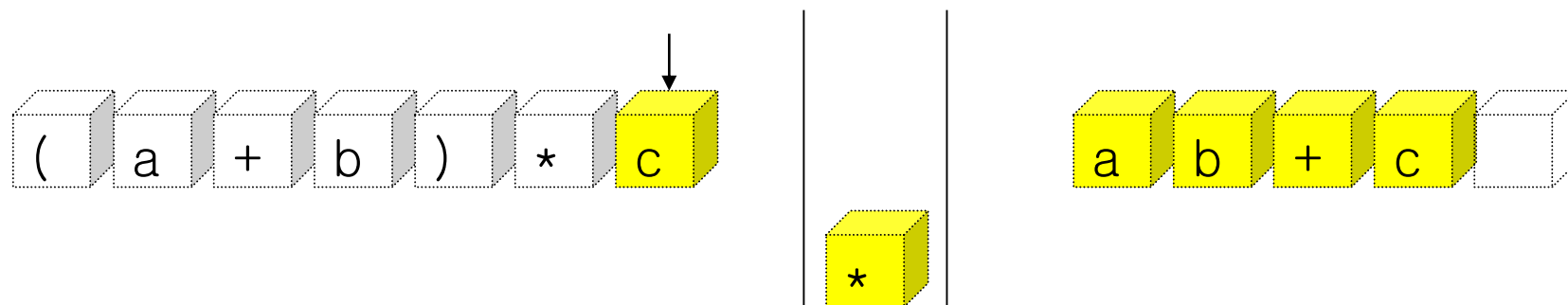
CSLAB

중위표기식 -> 후위표기식



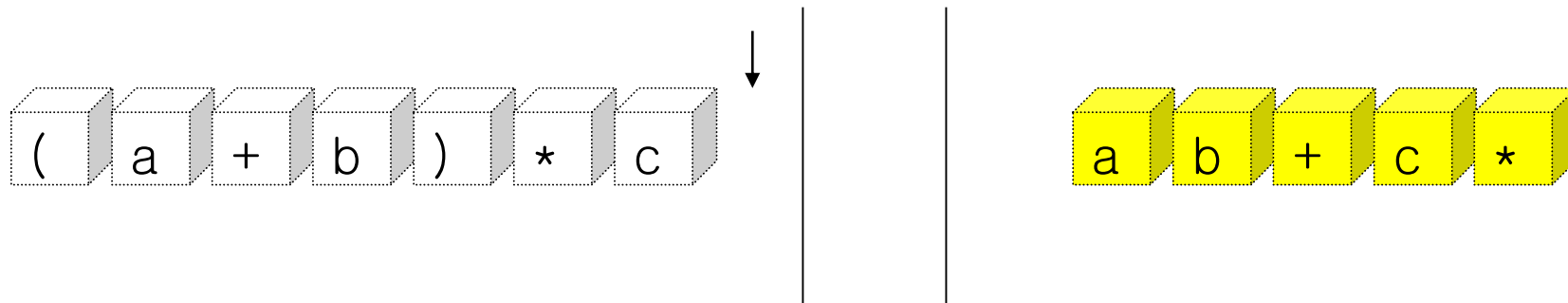
CSLAB

중위표기식 -> 후위표기식



CSLAB

중위표기식 -> 후위표기식



CSLAB

중위표기식 -> 후위표기식(pseudocode)

infix_to_postfix(exp)

스택 *s*를 생성하고 초기화

while (*exp*에 처리할 문자가 남아 있으면) **do**

ch ← 다음에 처리할 문자

switch (*ch*)

case 연산자:

while (*peek(s)*의 우선순위 \geq *ch*의 우선순위) **do**

e ← *pop(s)*

*e*를 출력

 push(*s*, *ch*);

break;

case 왼쪽 괄호:

 push(*s*, *ch*);

break;

case 오른쪽 괄호:

while(*e* \neq 왼쪽괄호) **do**

e ← *pop(s)*

*e*를 출력

break;

case 피연산자:

*ch*를 출력

break;

while (**not** is_empty(*s*)) **do**

e ← *pop(s)*

*e*를 출력

중위표기식 -> 후위표기식

- 개인 실습
 - 구조체

```
typedef struct {  
    char stack[MAX_STACK_SIZE];  
    int top;  
}StackType;
```

CSLAB

중위표기식 -> 후위표기식

- 주요함수

```
// 스택 초기화 함수
void init(StackType *s)
{
}

// 공백 상태 검출 함수
int is_empty(StackType *s)
{
}

// 포화 상태 검출 함수
int is_full(StackType *s)
{
}
```

중위표기식 -> 후위표기식

- 주요함수

// 삽입함수

```
void push(StackType *s, element item)
{
    if ( is_full(s) ) {
        return;
    }
    //
}
```

// 삭제함수

```
element pop(StackType *s)
{
    if ( is_empty(s) ) {
        exit(1);
    }
    //
}
```

// 피크함수

```
element peek(StackType *s)
{
    if ( is_empty(s) ) {
        fprintf(stderr, "스택 공백 에러\n"); exit(1);
    }
    //
}
```

중위표기식 -> 후위표기식

- 주요함수

```
// 연산자의 우선순위를 반환한다.  
int prec(char op)  
{  
    switch(op){  
  
    }  
    return -1;  
}
```

연산자 우선 순위:
 $\underbrace{‘(‘ ‘)’}_{0} < \underbrace{‘+’ ‘-’}_{1} < \underbrace{‘*’ ‘/’}_{2}$

중위표기식 -> 후위표기식

- 주요함수

```
void infix_to_postfix(char exp[])
{
    // 스택 생성, 초기화
    // 문자 처리
        // 연산자
        // 괄호
        // 피연산자
}
```

CSLAB

중위표기식 -> 후위표기식

- 메인함수

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_STACK_SIZE 20
int main()
{
    infix_to_postfix("(2+3)*4+9");
    return 0;
}
```

23+4*9+

CSLAB

제출

• 제출

- 개인실습 (#1)
 - 오늘 자정까지 제출 (~ 2020/5/15 23:59)
 - Infix to postfix conversion
- 과제
 - Lab9.docx
 - 다음주 목요일 자정까지 제출 (~2020/5/21 23:59)
 - Postfix evaluation

CSLAB

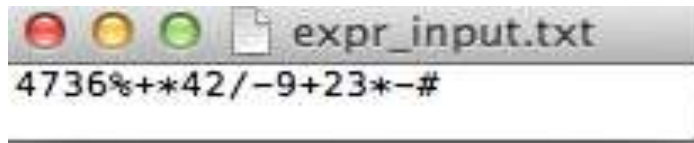
과제

- Lab9.docx

- 알고리즘

- 피연산자는 스택에 push
 - 연산자를 만나면 피연산자 두 개를 꺼내서 계산 후, 결과를 스택에 push

- Input



- output

```
converted postfix form : 4736%+*42/-9+23*~#
evaluation result : 41
```

CSLAB