

Stack

Hsuan-Tien Lin

Dept. of CSIE, NTU

March 31, 2020

Application: Expression Evaluation

Stack for Expression Evaluation

$$\underbrace{a/b - c + d * e - a * c}$$

- precedence: {*, /} first; {+, -} later
- steps

- $f = a/b$
- $g = f - c$
- $h = d * e$
- $i = g + h$
- $j = a * c$
- $l = i - j$

Postfix Notation

- same operand order, but put “operator” **after** needed operands
- can “operate” immediately when seeing operator
- no need to look beyond for precedence

Postfix from Infix (Usual) Notation

$\frac{\text{左}}{\text{右}}$ $\frac{\text{中}}{\text{右}}$

- infix:

3 / 4 - 5 + 6 * 7 - 8 * 9

- parenthesize:

$\left(\left(\left((3 / 4) - 5 \right) + (6 * 7) \right) - (8 * 9) \right)$

- for every triple in parentheses, switch orders

$\left(\left(\left((3 4 /) 5 - \right) (6 7 *) + \right) (8 9 *) - \right)$

- remove parentheses

3 4 / 5 - 6 7 * + 8 9 * -

difficult to parenthesize efficiently

Evaluate Postfix Expressions

$$34 / 5 - 67 * + 89 * -$$

(Handwritten annotations: a red arrow points to the expression; red underlines are under 34, 5, 67, 89, and the final minus sign; a red circle is around the division symbol; a vertical stack of colored scribbles is to the right.)

- how to evaluate? left-to-right, “operate” when see operator
 - 3, 4, / \Rightarrow 0.75
 - 0.75, 5, - \Rightarrow -4.25
 - -4.25, 6, 7, * \Rightarrow -4.25, 42 (note: -4.25 stored for latter use)
 - -4.25, 42, + \Rightarrow 37.75
 - 37.75, 8, 9, * \Rightarrow 37.75, 72 (note: 37.75 stored for latter use)
 - 37.75, 72, - \Rightarrow ...
- (Handwritten annotations: a red box encloses the second and third items; a red circle is around the first '-4.25' in the fourth item; a red box encloses the second and third items of the fifth item; a vertical stack of colored scribbles is to the right, with '37.75' written in green in the second cell.)*

stored where?

stack so closest operands will be considered first!

Stack Solution to Postfix Evaluation

Postfix Evaluation

for each *token* in the input **do**
 if *token* is a number
 push *token* to the stack)
 else if *token* is an operator
 sequentially pop operands a_{t-1}, \dots, a_0 from the stack
 push $token(a_0, a_1, a_{t-1})$ to the stack)
 end if
end for
return the top of stack

5 * []

matches closely with the definition of postfix notation

[] ? [] : []

Application: Expression Parsing

One-Pass Algorithm for Infix to Postfix

infix \Rightarrow postfix efficiently?

- at /, not sure of what to do (need later operands) so **store**

$$a/b - c + d * e - a * c$$

Handwritten annotations: A blue box around 'a/b', red dashes under 'c', 'd', 'e', and 'a', and a blue arrow pointing from the box to the '-' operator.

- at -, know that a / b can be a b / because - is of lower precedence

$$a/b - c + d * e - a * c$$

Handwritten annotations: 'a/b' is written below the first '-', and the first '-' is crossed out with a blue line.

- at +, know that ? - c can be ? c - because + is of same precedence but { -, + } is left-associative

$$a/b - c + d * e - a * c$$

Handwritten annotations: A blue box around 'a/b - c', a blue arrow pointing from the box to the '+' operator, and a blue '+' sign to the right.

- at *, not sure of what to do (need later operands) so **store**

$$a/b - c + d * e - a * c$$




Handwritten annotations: A blue box around 'd * e', a blue arrow pointing from the box to the '-' operator, and a blue arrow pointing from the box to the '+' operator.



stored where? **stack** so closest operators will be considered first!

Stack Solution to Infix-Postfix Translation

for each token in the input do
 if token is a number
 output token
 else if token is an operator
 while top of stack is of higher (or same) precedence **do**
 pop and output top of stack
 end while
 push token to the stack
 end if
end for

- here: infix to postfix with operator stack
—closest operators will be considered first
- recall: postfix evaluation with operand stack
—closest operands will be considered first
- mixing the two algorithms (say, use two stacks): simple calculator

$3 * 5 + 5566 - 1126$

Some More Hints on Infix-Postfix Translation

for each *token* in the input **do**

if *token* is a number

output *token*

else if *token* is an operator

while top of stack is of higher (or same) precedence **do**
pop and output top of stack

end while

push *token* to the stack

end if

end for

$3 * (5 + 7)$

- for left associativity and binary operators
 - right associativity? same precedence needs to wait
 - unary/trinary operator? same
- parentheses? highest priority
 - at '(', cannot pop anything from stack
—like seeing '*' while having '+' on the stack
 - at ')', can pop until '(' —like parentheses matching

