CSE 595: Advanced Topics in Computer Science
Presentation 4

Zeeshan Shaikh

Department of Computer Science, Stonybrook University

06/17/2021

Topics for today's presentation

- Expression Evaluation.

Problem 4: Expression Evaluation

Understanding the problem:

- Given: An arithmetic expression.
- Task: To evaluate the given arithmetic expression which can be in Infix, Prefix or Postfix Notation.
- Example: "$100 * (2 + 12)$", Expected Output $= 1400$

Challenges:

- Infix notations are how humans represent a problem.
- For computers, Infix Expressions are much more difficult to process as they require additional work to decide precedence.

Possible Approaches:

- Shunting Yard Algorithm by Edgar Dijkstra is a well known algorithm that is used to convert an infix notation to a postfix notation.
- We will go over that algorithm briefly and then, discuss a modified version of this algorithm that outputs the result of this evaluation.

Shunting Yard Algorithm

Why is this called as the "Shunting Yard" Algortihm?
That's because the algorithm represents a three-way railroad junction, similar to one you'd find in a railroad shunting yard.
Rules of the algorithm:

- If element is a number: Push to output
- If element is a function: Push to the operator stack
- If element is a an operator: Check for precedence
- If precedence $<$ precedence of the operator on top of the stack: Pop the operator on top and add to output.
- Remove any other remaining operators in the stack.

Visualization of the Shunting Yard Algorithm

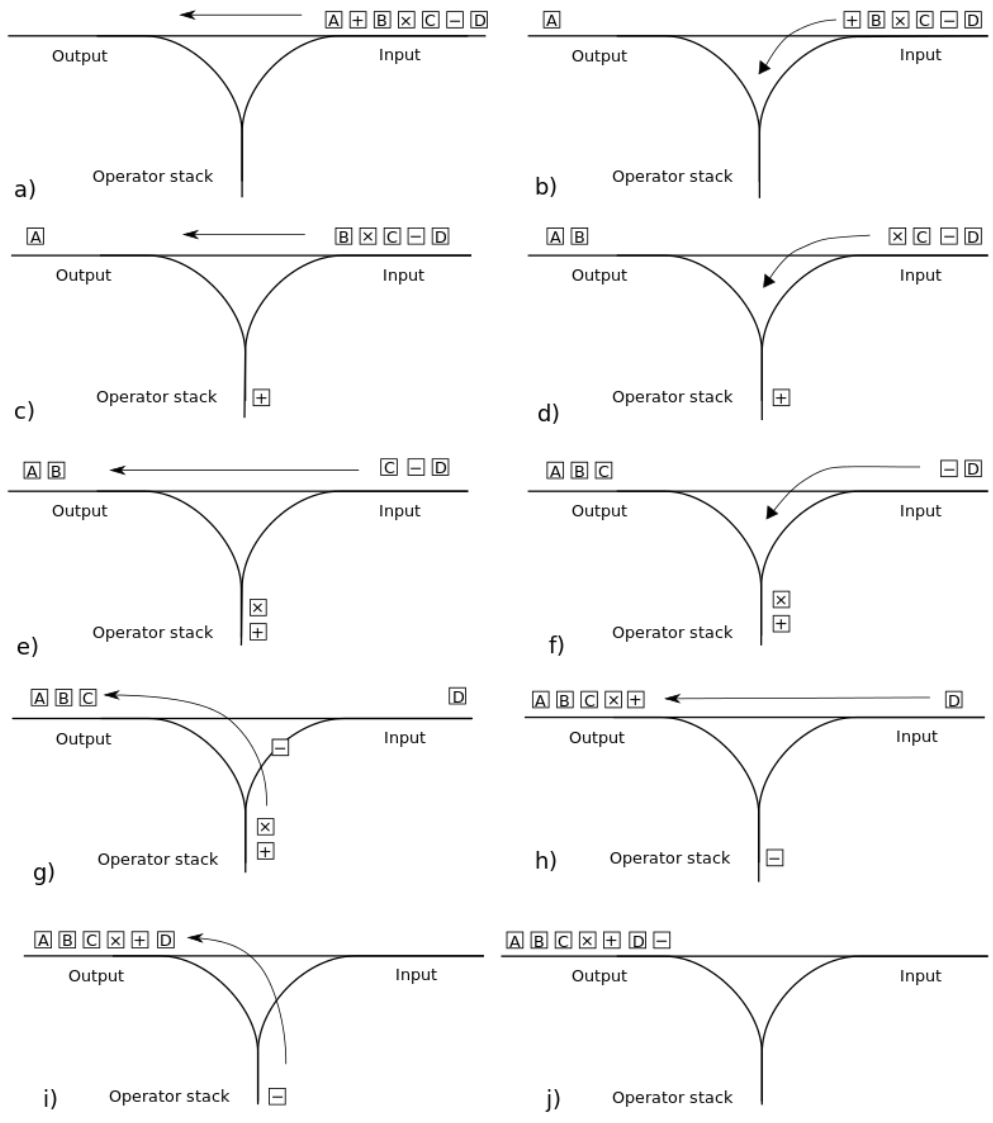

Figure 1: Shunting Yard Algorithm

Modified Shunting Yard Algorithm

Rules of the algorithm:

- If element is a number/variable: Push to value stack
- If element is a left parenthesis: Push to the operator stack
- If element is a right parenthesis; Repeat these steps till you reach a left parenthesis:
  - Pop an operator from operator stack
  - Pop two values from value stack
  - Apply operator to the values and push results to value stack.
- Pop the left parenthesis
- If element is an operator:
  - If the operator on top of the stack has $>=$ precedence than the current operator:
    - Pop an operator from operator stack
    - Pop two values from value stack
    - Apply operator to the values and push results to value stack.
  - Push this operator to the operator stack

Complexity Analysis of the modified Shunting Yard Algorithm.

Complexity Analysis		
Algorithm	Time	Space
Modified Shunting Yard	$O(N)$	$O(N)$

Visualization of the Modified Shunting Yard Algorithm

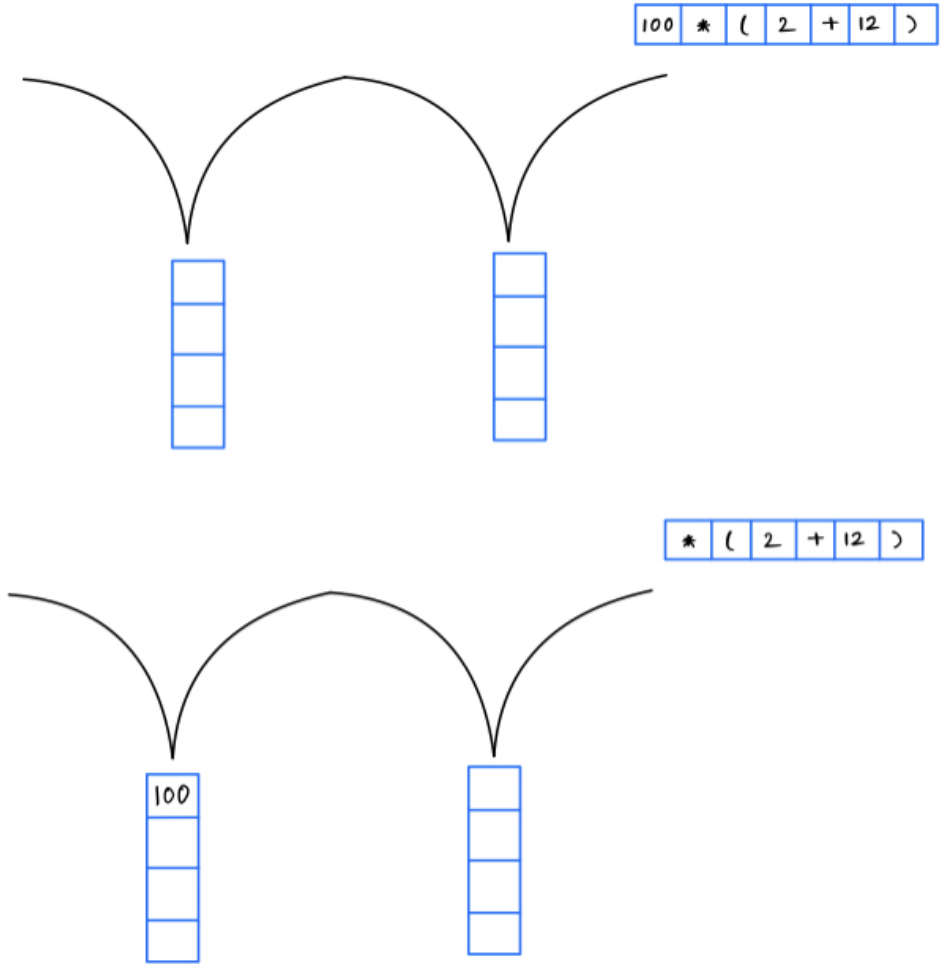

Figure 2: Modified Shunting Yard Algorithm

Visualization of the Modified Shunting Yard Algorithm

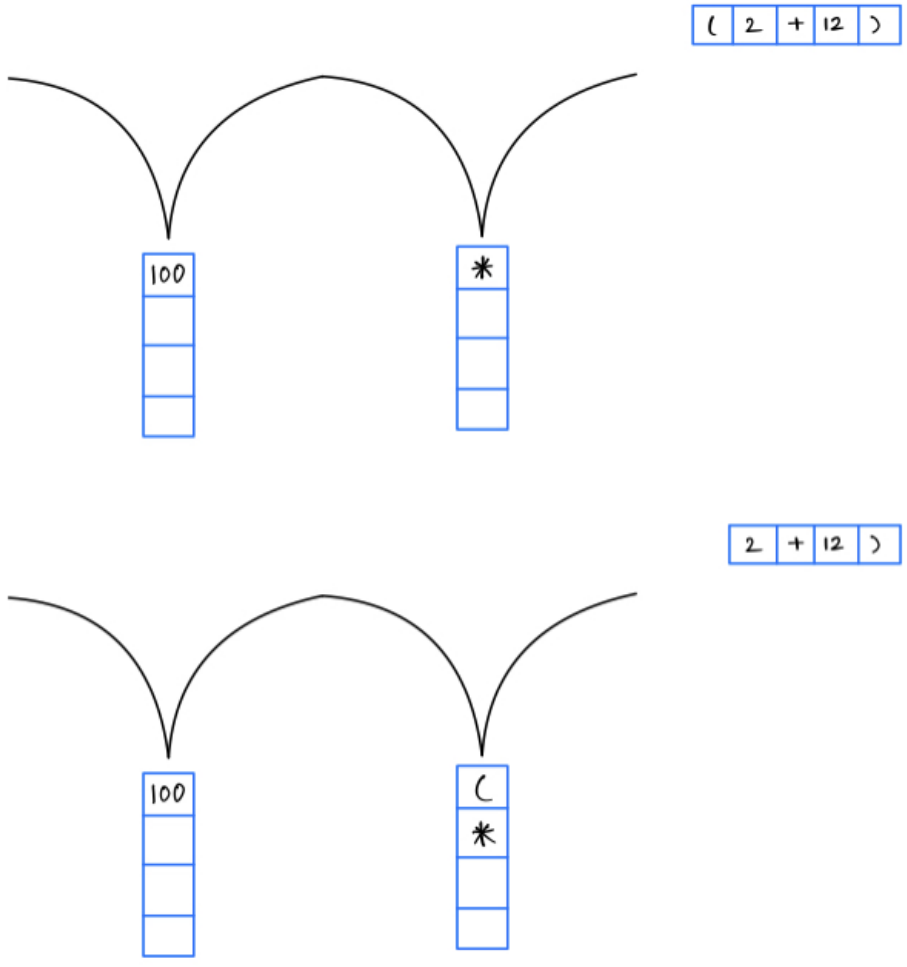

Figure 3: Modified Shunting Yard Algorithm

Visualization of the Modified Shunting Yard Algorithm

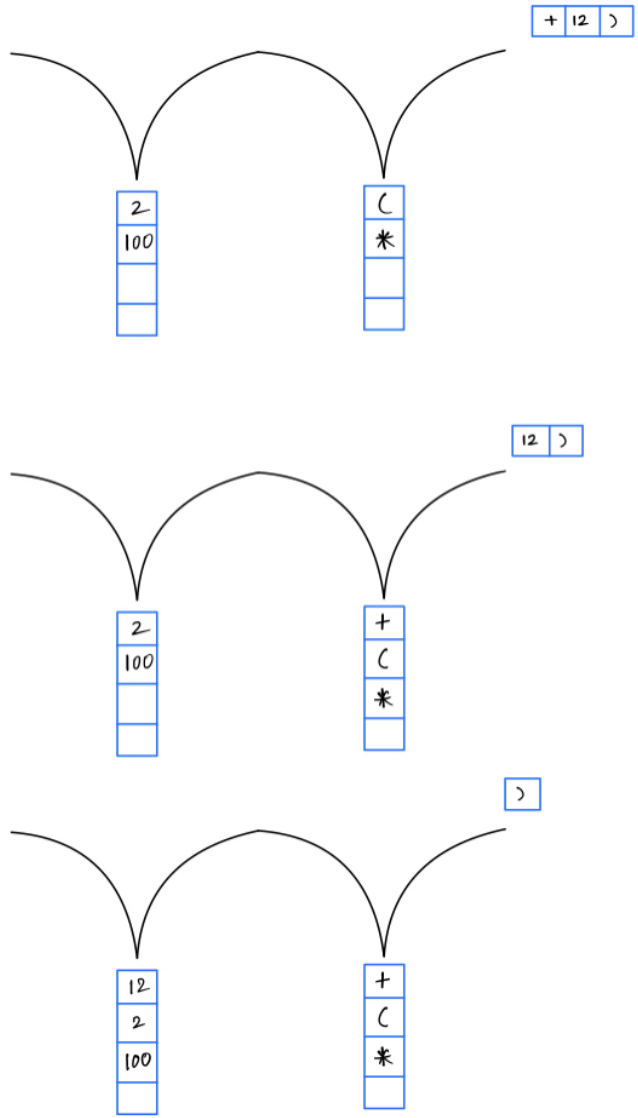

Figure 4: Modified Shunting Yard Algorithm

Visualization of the Modified Shunting Yard Algorithm

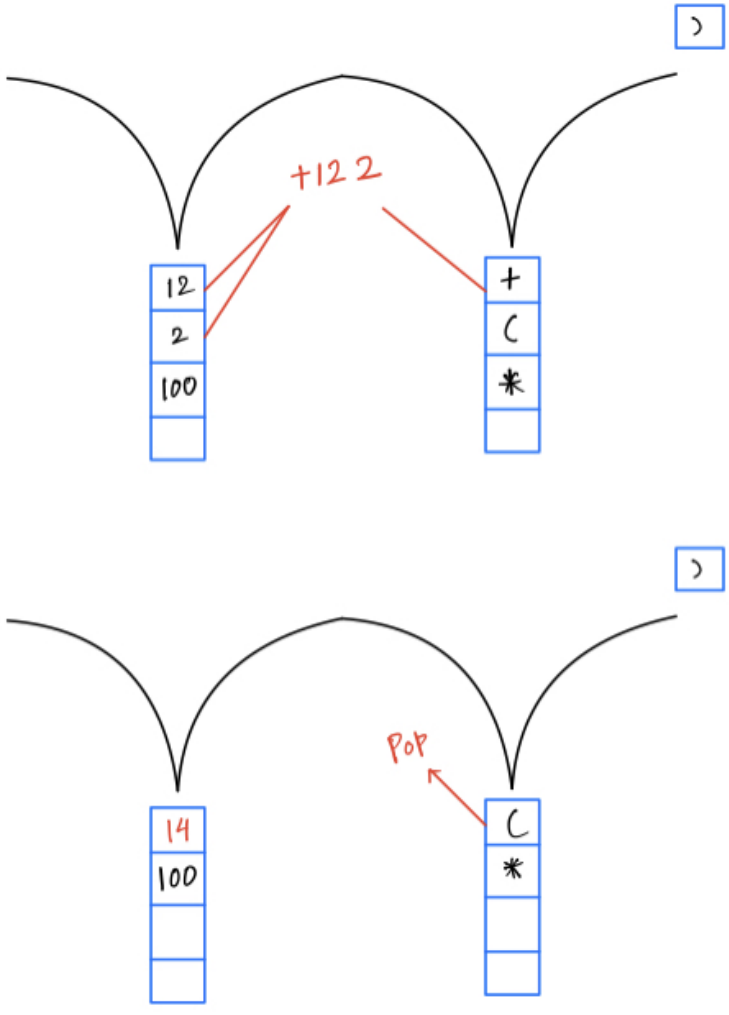

Figure 5: Modified Shunting Yard Algorithm

Visualization of the Modified Shunting Yard Algorithm

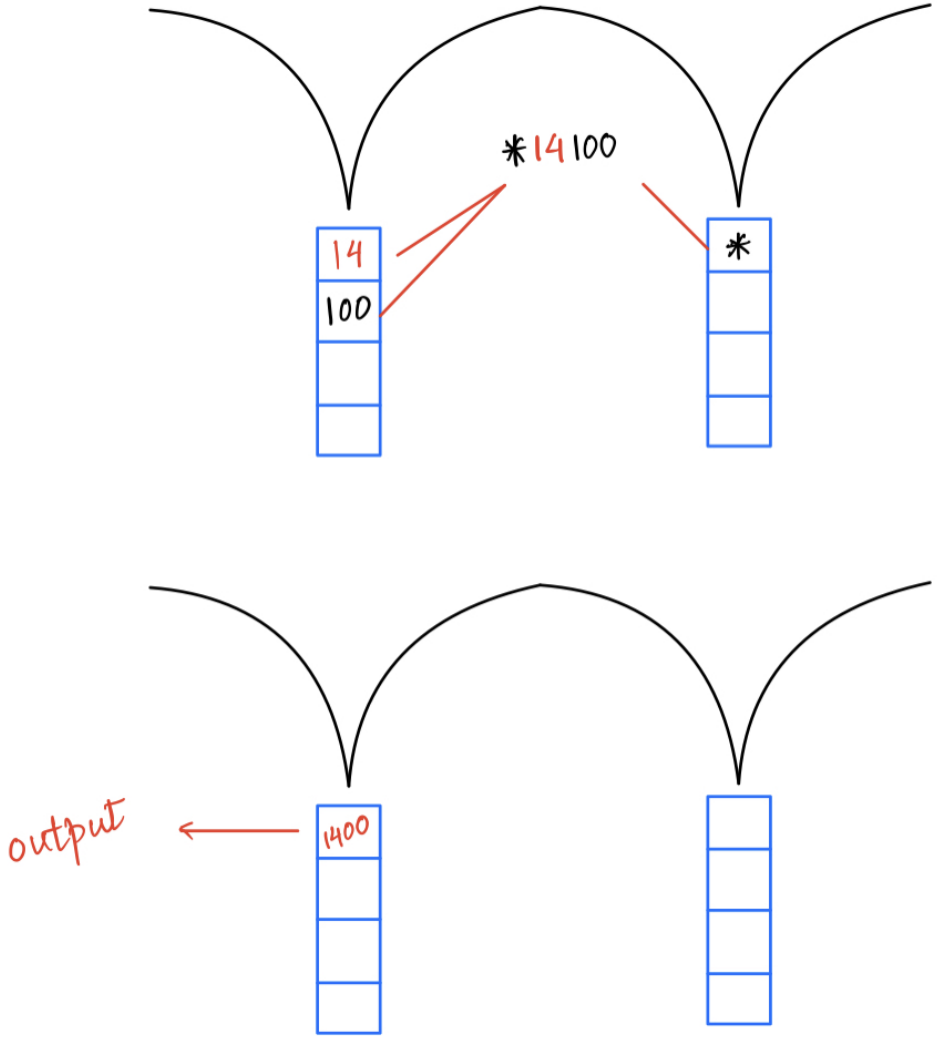

Figure 6: Modified Shunting Yard Algorithm

Thankyou!