# **Thompsons Construction Documentation**

## Shunting Yard Algorithm

(Reference Wikipedia <https://en.wikipedia.org/wiki/Shunting-yard_algorithm>)

In [computer science](https://en.wikipedia.org/wiki/Computer_science), the **shunting-yard algorithm** is a method for parsing mathematical expressions specified in [infix notation](https://en.wikipedia.org/wiki/Infix_notation). It can produce either a postfix notation string, also known as [Reverse Polish notation](https://en.wikipedia.org/wiki/Reverse_Polish_notation) (RPN), or an [abstract syntax tree](https://en.wikipedia.org/wiki/Abstract_syntax_tree) (AST). The [algorithm](https://en.wikipedia.org/wiki/Algorithm) was invented by [Edsger Dijkstra](https://en.wikipedia.org/wiki/Edsger_Dijkstra" \o "Edsger Dijkstra) and named the "shunting yard" algorithm because its operation resembles that of a [railroad shunting yard](https://en.wikipedia.org/wiki/Classification_yard).

Converting an infix regular expression to postfix using the Shunting Yard Algorithm

1. Input: 3 + 4
2. Push 3 to the output queue (whenever a number is read it is pushed to the output)
3. Push + (or its ID) onto the operator stack
4. Push 4 to the output queue
5. After reading the expression, pop the operators off the stack and add them to the output.

In this case there is only one, "+".

1. Output: 3 4 +

This already shows a couple of rules:

* All numbers are pushed to the output when they are read.
* At the end of reading the expression, pop all operators off the stack and onto the output.

## Shunting Example

(Reference The Oxford Math Centre <http://www.oxfordmathcenter.com/drupal7/node/628>)

A \* B + C becomes A B \* C +

The order in which the operators appear is not reversed. When the '+' is read, it has lower precedence than the '\*', so the '\*' must be printed first.

We will show this in a table with three columns. The first will show the symbol currently being read. The second will show what is on the stack and the third will show the current contents of the postfix string. The stack will be written from left to right with the 'bottom' of the stack to the left.

A \* B + C --> A B \* C +

Current Symbol Operator Stack Postfix String

1 A A

2 \* \* A

3 B \* A B

4 + + A B \* (pop and print \* before pushing +)

5 C + A B \* C

6 A B \* C +

The rule used in lines 1, 3 and 5 is to print an operand when it is read. The rule for line 2 is to push an operator onto the stack if it is empty. The rule for line 4 is if the operator on the top of the stack has higher precedence than the one being read, pop and print the one on top and then push the new operator on. The rule for line 6 is that when the end of the expression has been reached, pop the operators on the stack one at a time and print them.

## Regular Expressions

(References Wikipedia <https://en.wikipedia.org/wiki/Regular_expression>)

A regular expression, regex or regexp (sometimes called a rational expression) is a sequence of characters that define a *search pattern*. Usually this pattern is used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation. It is a technique that developed in theoretical computer science and formal language theory.

A regular expression, often called a pattern, is an expression used to specify a set of strings required for a particular purpose. A simple way to specify a finite set of strings is to list its elements or members.

A quantifier after a token (such as a character) or group specifies how often that a preceding element can occur. The most common quantifiers are the [question mark](https://en.wikipedia.org/wiki/Question_mark) ?, the asterisk \* (derived from the Kleene star), and the plus sign + (Kleene plus).

|  |  |
| --- | --- |
| ? | The question mark indicates *zero or one* occurrences of the preceding element. For example, colou?r matches both "color" and "colour". |
| \* | The asterisk indicates *zero or more* occurrences of the preceding element. For example, ab\*c matches "ac", "abc", "abbc", "abbbc", and so on. |
| + | The plus sign indicates *one or more* occurrences of the preceding element. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac". |
| **{n}**[[18]](https://en.wikipedia.org/wiki/Regular_expression#cite_note-grep-18) | The preceding item is matched exactly *n* times. |
| **{min,}**[[18]](https://en.wikipedia.org/wiki/Regular_expression#cite_note-grep-18) | The preceding item is matched *min* or more times. |
| **{min,max}**[[18]](https://en.wikipedia.org/wiki/Regular_expression#cite_note-grep-18) | The preceding item is matched at least *min* times, but not more than *max* times. |