# Алгоритм поиска подстрок с помощью конечных автоматов

Данный проект рассчитан на пользователей, имеющих хотя бы небольшое представление о том, что такое конечные автоматы. Поэтому не будем углубляться в разбор их самих, а сосредоточимся непосредственно на конечных автоматах, применяемых для нашего алгоритма.

# Основные сведения

Алгоритм поиска подстрок с помощью конечных автоматов начинается, непосредственно, с построения конечного автомата по имеющемуся образцу (искомой подстроке). Эффективность и привлекательность этого алгоритма состоит в том, что при последующем анализе исходной строки каждый её символ обрабатывается один единственный раз, что обеспечивает линейную сложность непосредственно поиска подстрок, что обеспечивает большую эффективность при больших объёмах входных данных. Однако время построения самого автомата зависит от используемого алфавита и, при большом его объёме, может быть достаточно значительным.

## Автоматы поиска подстрок

Для каждого образца существует свой автомат поиска, так что его необходимо сконструировать на этапе предварительной обработки, чтобы затем его можно было использовать для поиска необходимых нам подстрок.

Наиболее наглядно объяснить процесс построения можно на конкретном примере.

Возьмём за образец строку **ababaca**. Алфавит примем состоящим лишь из символов **a, b, c.**