1. Алфавит I конечного автомата — это множество допустимых входных символов, которые могут поступать на вход конечного автомата. Каждый символ из алфавита I может вызывать переход автомата из одного состояния в другое или вызывать действие внутри автомата.

* Символ λ (лямбда) обозначает пустую строку, то есть последовательность символов длиной 0. В регулярных выражениях и конечных автоматах он используется для обозначения переходов без входного символа.
* Символ \* обозначает операцию звезды Клини, которая обозначает повторение нуля или более раз символа или подвыражения, на которое он применяется.
* Символ + обозначает операцию плюс, которая обозначает повторение одного или более раз символа или подвыражения, на которое он применяется.
* I обозначает алфавит (набор символов), которые могут использоваться в регулярном выражении или конечном автомате.

1. Язык L(I) над алфавитом I — это множество строк, которые можно составить, используя символы из алфавита I.
2. Формальная грамматика G - это четверка G = (V, T, P, S), где:

* V - конечное множество нетерминальных символов (переменных)
* T - конечное множество терминальных символов (констант, имен и т.д.)
* P - множество правил вывода, описывающих, как нетерминальные символы могут быть заменены на последовательности терминальных и нетерминальных символов. Каждое правило вывода имеет вид A -> β, где A - нетерминальный символ, а β - последовательность терминальных и/или нетерминальных символов.
* S - стартовый символ, или начальный нетерминальный символ, от которого начинается вывод цепочек терминальных символов.

Регулярная грамматика — это формальная грамматика, задающая регулярный язык, то есть язык, который может быть распознан конечным автоматом.

Регулярное множество - это множество строк, которые могут быть сгенерированы с помощью регулярной грамматики или распознаны с помощью конечного автомата.

Регулярный язык - это множество строк, которые могут быть порождены из пустой строки с помощью конечного автомата или регулярной грамматики

Лексический анализ (англ. lexical analysis или scanning) - это первый этап компиляции программы или интерпретации скрипта, заключающийся в преобразовании входной последовательности символов (обычно текст программы) в последовательность лексем или токенов, которые представляют собой минимальные синтаксические единицы программы. Лексический анализатор (лексер) преобразует последовательность символов в поток лексем, идентифицируя каждую лексему и присваивая ей тип и значение, если оно есть.

Последовательный лексический анализатор (sequential lexical analyzer) анализирует входную последовательность символов поочередно, последовательно применяя к ним правила лексического анализа. Такой анализатор имеет простую реализацию, но может быть неэффективным, если входная последовательность достаточно длинная.

Параллельный лексический анализатор (parallel lexical analyzer) анализирует входную последовательность символов параллельно, используя несколько лексических анализаторов, каждый из которых работает со своим набором правил. Такой анализатор может быть более эффективным, поскольку обработка входной последовательности происходит параллельно. Однако реализация такого анализатора может быть сложнее, чем у последовательного.

Детерминированный автомат (DFA) и недетерминированный автомат (NFA) - это два типа конечных автоматов, используемых для распознавания регулярных языков. Главное отличие между ними заключается в том, что DFA имеет один переход для каждой пары состояние-входной символ, в то время как NFA может иметь несколько переходов для этой пары.

Другими словами, в DFA каждому состоянию и входному символу соответствует один переход, в то время как в NFA для данного состояния и входного символа может быть несколько переходов в разные состояния. Поэтому DFA является более простым в реализации и обладает более быстрой обработкой, но NFA более выразителен и может обрабатывать некоторые классы языков, которые не могут быть обработаны DFA.