lab2 Амбросий Николай Евгеньевич ИУ9-51Б

1 минимальный ДКА

Имеем регулярное выражение:

$$((a^*b^*c^*)^*ab(a^*b^*c^*)^*bc(a|b|c)^*) \mid ((a|b|c)^*bc(a^*b^*c^*)^*ab(a|bc|cc|bb)^*) \mid abc$$

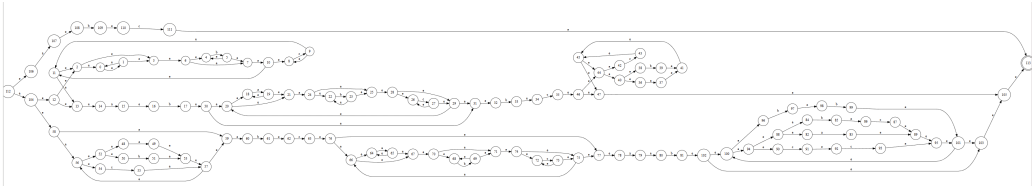
Преобразуем его в вид, удобный для парсинга. Явно обозначим конкатенацию символом «$\cdot$»:

$$\begin{gathered}((a^* \cdot b^* \cdot c^*)^* \cdot a \cdot b \cdot (a^* \cdot b^* \cdot c^*)^* \cdot b \cdot c \cdot (a|b|c)^*) \mid \\ ((a|b|c)^* \cdot b \cdot c \cdot (a^* \cdot b^* \cdot c^*)^* \cdot a \cdot b \cdot (a|b \cdot c|c \cdot c|b \cdot b)^*) \mid \\ a \cdot b \cdot c\end{gathered}$$

Преобразуем полученное выражение в Обратную Польскую Нотацию:
$a * b * .c * . * a.b.a * b * .c * . * .b.c.ab|c| * .ab|c| * b.c.a * b * .c * . * .a.b.abc.|cc.|bb.| * .|ab.c.|$

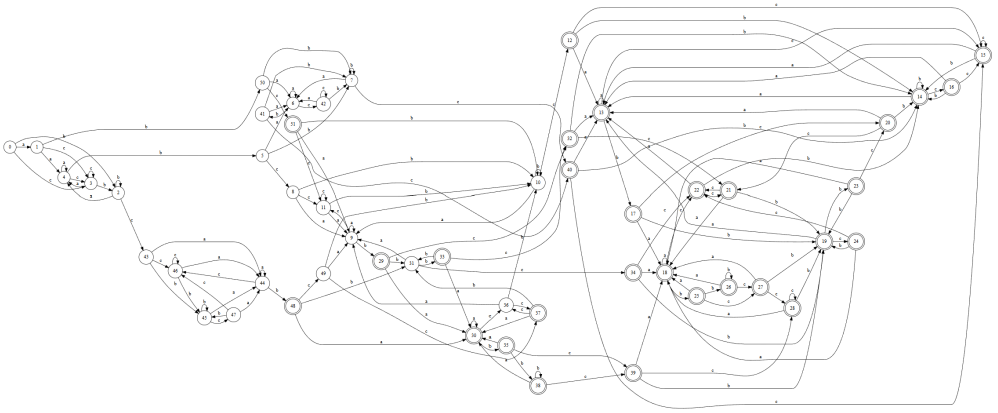
2 Построение НКА (Алгоритм Томпсона)

Применим метод Томпсона к обратной польской записи и НКА.

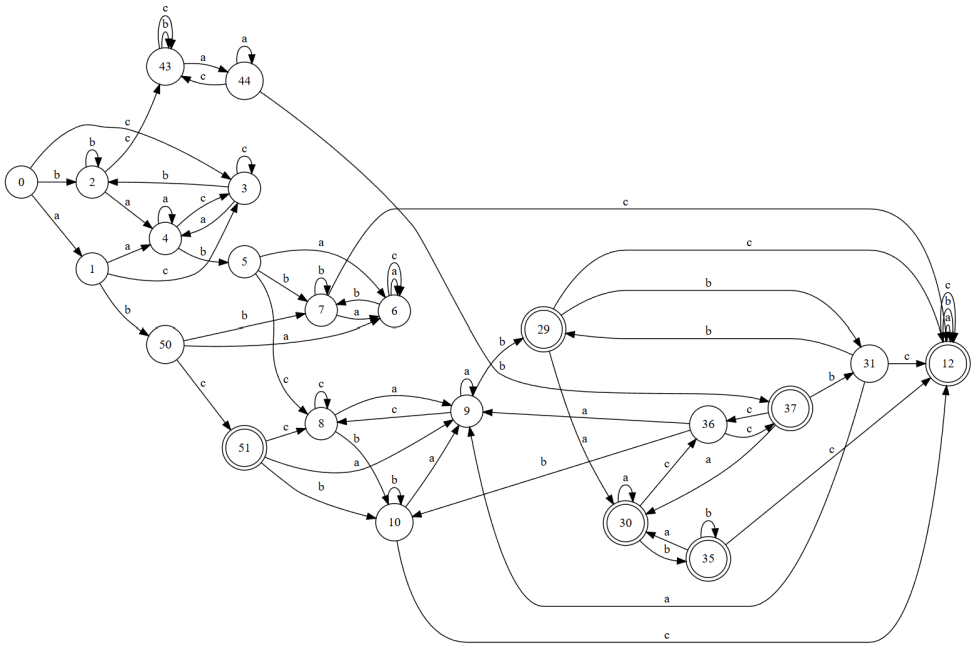


3 Детерминизация и минимизация

Выполним детерминизацию полученного НКА



А затем минимизацию полученного ДКА.



4 малый НКА

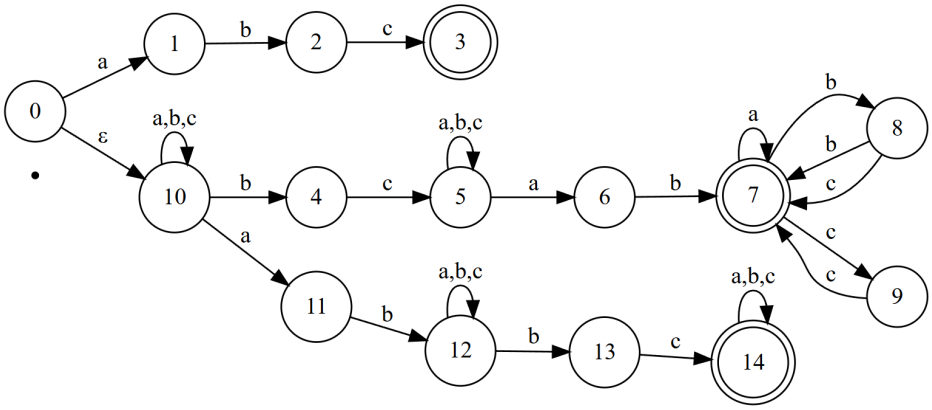
Упростим исходное регулярное выражение.

Заметим, что конструкция $(a^*b^*c^*)^*$ при условии алфавита $\Sigma = \{a, b, c\}$ эквивалентна $(a|b|c)^*$.

Упрощенное выражение принимает вид:

$$((a|b|c)^*ab(a|b|c)^*bc(a|b|c)^*) \mid ((a|b|c)^*bc(a|b|c)^*ab(a|bc|cc|bb)^*) \mid abc$$

построим по нему малый НКА



5 Расширенные регулярные выражения

Вынесем проверку алфавита $\{a, b, c\}$ в lookahead. Тогда конструкции $(a|b|c)^*$ можно заменить на $.^*$.

$(?=[abc]^*\$)((.^*ab.^*bc.^*)|(.^*bc.^*ab(a|bc|cc|bb)^*)|abc)\$$

6 Построение ПКА

Наш язык L представляет собой объединение трех языков: $L = L_1 \cup L_2 \cup L_3$.

Соответственно И недетерминизм надо ввести в каком-нибудь L_i. У L_1 и L_3 структура линейна и проста, вводить распараллеливание негде

Остается L_2. В нем префиксная часть $.^*bc.^*ab$ также линейна. Единственное место, допускающее параллельную обработку — это суффикс: $(a \mid bc \mid cc \mid bb)^*$

Нас интересуют составные подслова bc и bb. Можно разделить обработку на два потока:

1. Поток 1: После нахождения b просто продолжает чтение, ожидая новые символы согласно структуре.

2. Поток 2: Проверяет, является ли текущая последовательность допустимым подсловом (bc или bb) в контексте суффикса.