

1 Задание

Дано следующее академическое регулярное выражение:

$$((aa|ab|cc)^*aba(aaa|bcc)^*)((abac|(cc)^*)(b|ca))^*$$

По имеющемуся академическому регулярному выражению построить

- минимальный ДКА, распознающий его язык (минимальность обосновать таблицей классов эквивалентности)
- возможно малый НКА, распознающий его язык. Возможно малый переключающийся (с конъюнкцией) КА, распознающий его язык. Частично обосновать таблицами множеств классов эквивалентности.
- расширенное регулярное выражение, распознающее тот же язык. В расширенном выражении можно использовать:
 - wildcard-операцию для замены произвольного алфавита;
 - положительную итерацию τ^+ и опцию $\tau^+.\tau^+ = \tau\tau^*$, $\tau^? = (\tau|\varepsilon)$;
 - операции предпросмотра $\tau_0(? = \tau_1)\tau_2 \equiv \tau_0((\tau_1.^*) \cap \tau_2)$ и ретроспективной проверки $\tau_0(? \leq \tau_1)\tau_2 \equiv (\tau_0 \cup (\tau_1.^*))\tau_2$, а также их отрицательные версии $\tau_0(? \neq \tau_1)\tau_2 \equiv \tau_0(\overline{(\tau_1.^*)} \cap \tau_2)$ и $\tau_0(? < \tau_1)\tau_2 \equiv (\tau_0 \cap \overline{(\tau_1.^*)})\tau_2$;
 - классы букв $[c_1 \dots c_k] \equiv (c_1|c_2|\dots|c_k)$ и их дополнения $[\hat{c}_1 \dots \hat{c}_k]$;
 - (обязательно) маркеры начала и конца выражения \wedge и $\$$.

Провести автоматическое тестирование предполагаемой эквивалентности построенных распознавателей. Тем самым необходимо построить алгоритмы, определяющие принадлежность слова языку академического регулярного выражения, ДКА, НКА и ПКА.

Эквивалентность расширенного регулярного выражения тестировать не обязательно (на дополнительный балл). Если не делать, то для расширенного регулярного выражения достаточно описать содержательно (полуформально), почему оно должно распознавать тот же самый язык.

Требуется только фазз-тестирование эквивалентности: строится случайное слово ω и проверяется, принадлежит ли оно языкам регулярного выражения, ДКА, НКА и ПКА согласованно.

2 Минимальный ДКА

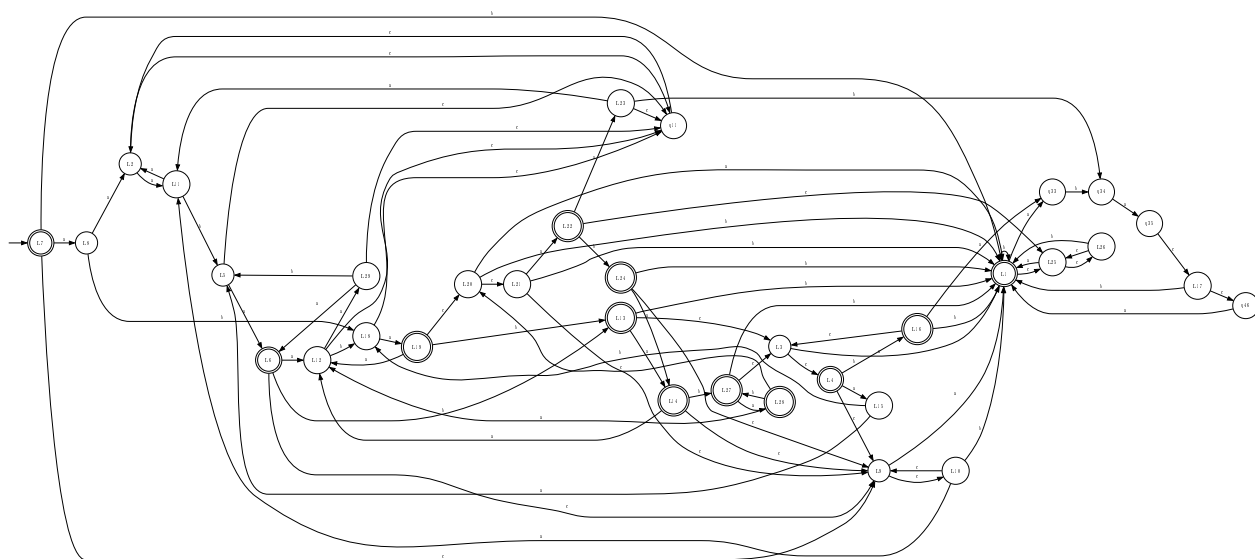


Рис. 1: Минимальный ДКА

Таблица классов эквивалентности:

Таблица 1: Таблица эквивалентности (строки - префиксы, столбцы - суффиксы)

	ε	aaa	aba	cc	ba	a	b	bacb	c	acb	cb	aa	saba	ccb
ε	+	-	+	-	-	-	+	-	-	-	-	-	-	+
a	-	-	-	-	+	-	-	+	-	-	-	-	-	-
b	+	-	-	-	-	-	+	-	-	-	-	-	-	+
c	-	-	-	-	-	+	-	-	-	-	+	-	+	-
aa	-	-	+	-	-	-	-	-	-	-	-	-	-	-
ab	-	-	+	-	-	+	-	-	-	+	-	-	-	-
ac	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ba	-	-	-	-	-	-	-	+	-	-	-	-	-	-
bc	-	-	-	-	-	+	-	-	-	-	+	-	-	-
cc	-	-	+	-	-	-	+	-	-	-	-	-	-	+
aaa	-	-	-	-	+	-	-	-	-	-	-	-	-	-
aac	-	-	-	-	-	-	-	-	-	-	-	-	+	-
aba	+	+	+	-	+	-	+	-	-	-	+	-	-	+
bab	-	-	-	-	-	-	-	-	-	+	-	-	-	-
bcc	-	-	-	-	-	-	+	-	-	-	-	-	-	+
aaab	-	-	+	-	-	+	-	-	-	-	-	-	-	-
abaa	-	-	+	-	+	-	-	+	-	-	-	+	-	-
abab	+	-	+	+	-	+	+	-	-	-	-	-	-	+
abac	-	-	-	-	-	+	+	-	-	-	+	-	+	-
baba	-	-	-	-	-	-	-	-	-	-	+	-	-	-
aaaba	+	+	+	-	+	-	+	-	-	-	-	-	-	+
abaaa	-	-	+	-	+	+	-	-	-	-	-	-	-	-
ababa	+	+	+	-	+	-	+	+	-	-	-	-	-	+
ababc	-	-	-	-	-	+	-	-	+	-	+	-	+	-
abacc	-	-	+	-	-	+	+	-	-	-	-	-	-	+
babac	-	-	-	-	-	-	+	-	-	-	-	-	-	-
ababab	+	-	+	+	-	+	+	-	-	+	-	-	-	+
ababcc	+	+	+	-	-	-	+	-	-	-	-	-	-	+
abacca	+	-	-	-	+	-	+	-	-	-	-	-	-	+
babacc	-	-	-	-	-	+	-	-	-	-	-	-	-	-
abababa	+	+	+	-	+	-	+	+	-	-	+	-	-	+
ababcca	-	-	-	-	+	-	-	+	-	-	-	+	-	-
ababccb	+	-	-	+	-	-	+	-	-	-	-	-	-	+
abacca	-	-	+	-	-	-	-	+	-	-	-	-	-	-
abaccab	+	-	+	-	-	+	+	-	-	-	-	-	-	+

3 Меньший НКА

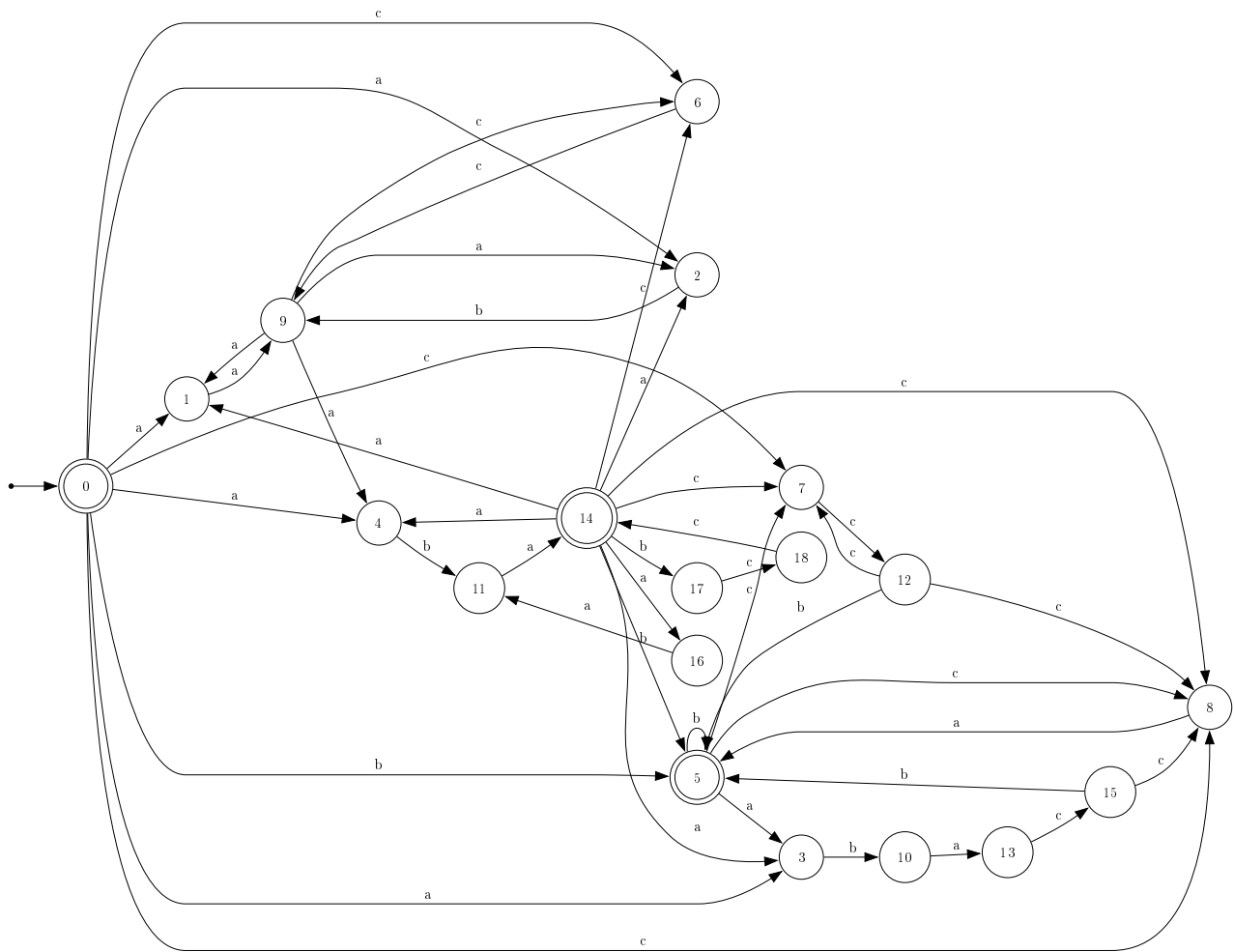


Рис. 2: Редуцированный автомат Глушкова

Таблица эквивалентности:

	ε	ba	cb	cc	acb
ε	+	-	-	-	-
a	-	+	-	-	-
aba	+	+	+	-	-
abab	+	-	-	+	-
ababab	+	-	-	+	+

4 Переключающийся автомат

Переключающийся автомат будет совпадать с НКА, так как нет нетривиальных инвариантов, которые можно будет проверять параллельно на всей строке. Проверка наличия подстрок aba или (b|ca) уже присутствует в НКА и является локально-последовательной.

5 Расширенное регулярное выражение

Исходное регулярное выражение:

$$((aa|ab|cc)^*aba(aaa|bcc)^*)((abac|(cc)^*)(b|ca))^*$$

Заметим, что подстроку $(aa|ab|cc)$ можно заменить на $(a[ab]|cc)$, а $[ab]$ можно поменять на дополнение $[\hat{c}]$. Получим в итоге такое выражение:

$$((a[\hat{c}]|cc)^*aba(aaa|bcc)^*)((abac|(cc)^*)(b|ca))^*$$

Далее, добавляем символы начала и конца выражения и получаем расширенное регулярное выражение:

$$^((a[\hat{c}]|cc)^*aba(aaa|bcc)^*)((abac|(cc)^*)(b|ca))^*\$$$

- wildcard-операцию применить нигде не получится.
- Положительная итерация и опция не имеют смысла потому, что у нас нет ситуации $\tau\tau^*$
- Операции предпросмотра, ретроспективной проверки и их отрицательные версии тоже не имеют смысла, потому что ничего не упростят, а только усложнят регулярку.