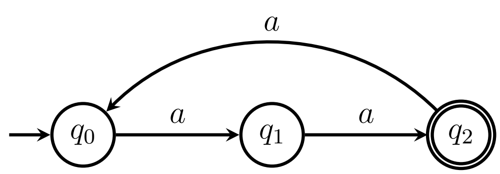
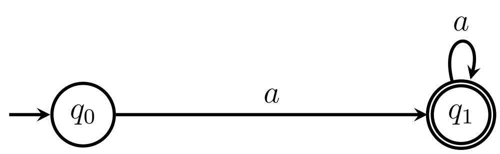
# Формальные языки

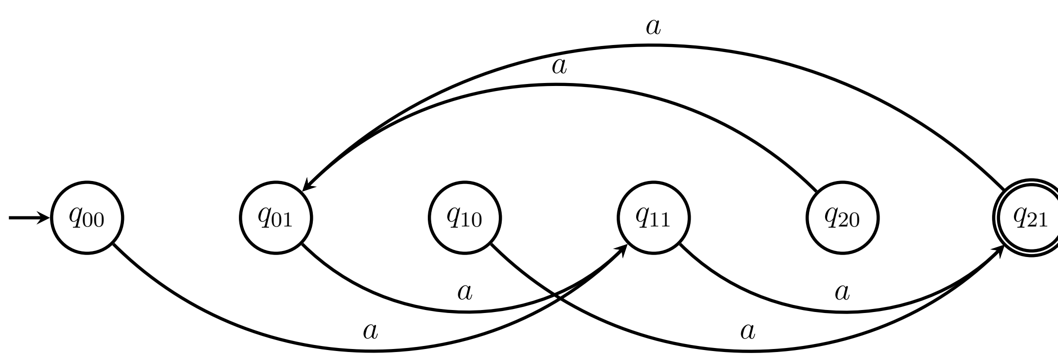
## домашнее задание до 23:59 05.03

1. Доказать или опровергнуть утверждение: произведение двух минимальных автоматов всегда дает минимальный автомат (рассмотреть случаи для пересечения, объединения и разности языков).
   Решение:
   1)Пересечение: Нет




Тогда их произведение:



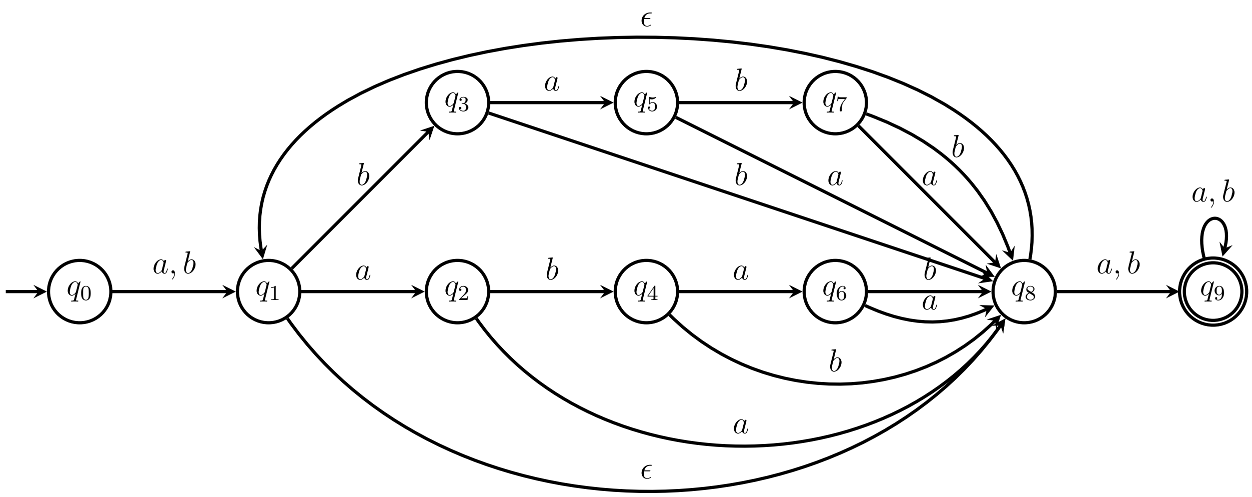

Первый принимает все строки длины сравнимой с 2 по модулю 3, а второй все слова длиной больше 1. Тогда произведение эквивалентно 1. Но как видно в нем 4 осмысленные вершины, а в минимальном 3.
2)Объединение:
Возьмем предыдущий пример. Тогда автомат произведения просто эквивалентен второму. Но в произведении опять 4 вершины.
3)Разность:
Изменим немного первый автомат. Пусть q0 будет тоже принимающей. Тогда он принимает все остатки кроме 1. Тогда произведение будет принимать толко остаток 1. Но оно состоит из 4 вершин. А минимальный из 3(это будет первый автомат если q1 сделать принимающей).

2. Для регулярного выражения:

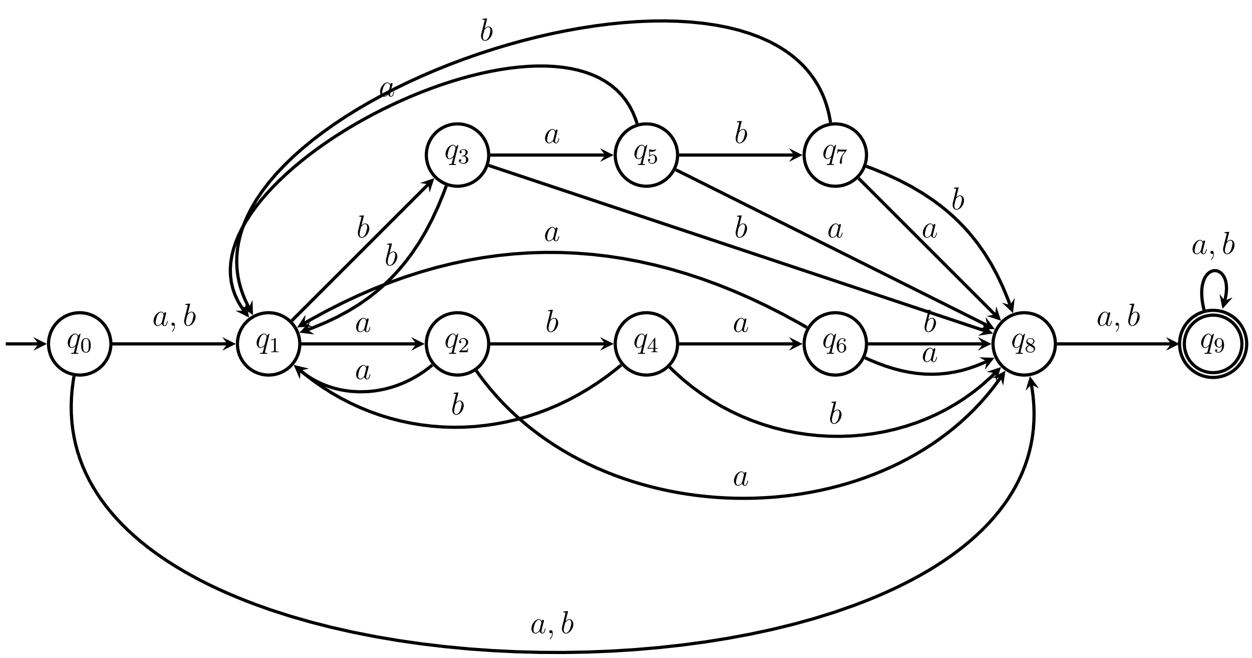
$$(a \mid b)^+(aa \mid bb \mid abab \mid baba)^*(a \mid b)^+$$

Построить эквивалентные:

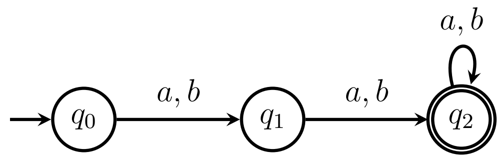
(a) Недетерминированный конечный автомат



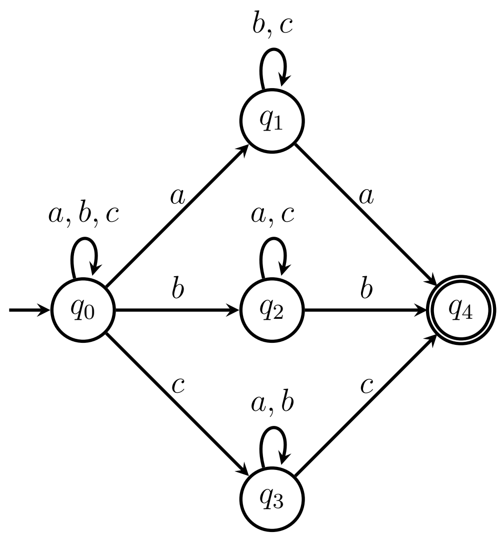

(b) Недетерминированный конечный автомат без $\varepsilon$-переходов



(c) Минимальный полный детерминированный конечный автомат
Детерминировать и минимизировать я его конечно не буду. На самом деле это регулярное выражение принимает любые слова длиной больше 1. Так что вот:




3. Построить регулярное выражение, распознающее тот же язык, что и автомат:



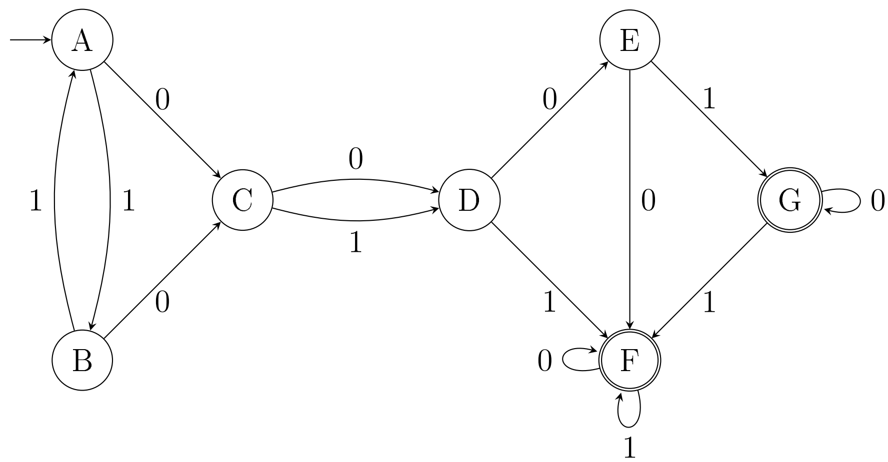

Решение:
$(a \mid b \mid c)^*((a(b \mid c)^*a) \mid (b(a \mid c)^*b) \mid (c(a \mid b)^*c))$

4. Определить, является ли автоматным язык $\{\omega\omega^r \mid \omega \in \{0,1\}^*\}$. Если является — построить автомат, иначе — доказать.
Решение:
Зафиксируем n. и возьмем например w $= 0^n110^n$. Ну тогда y из леммы всегда состоит из 0. И тогда получившееся слово например для k=2(из леммы) будет несимметричным.

5. Определить, является ли автоматным язык $\{uaav \mid u, v \in \{a,b\}^*, |u|_b \geq |v|_a\}$. Если является — построить автомат, иначе — доказать.
Решение:
Зафиксируем n. Возьмем w $= b^naa(ba)^n$. Тогда $y$ состоит из букв b. И если его выкинуть, то слово не будет принадлежать языку. То есть лемма не работает для k $= 0$.

# Пример применения алгоритма минимизации

Минимизируем данный автомат:




Автомат полный, в нем нет недостижимых вершин — продолжаем.

Строим обратное $\delta$ отображение.

| $\delta^{-1}$ | 0 | 1 |
|---|---|---|
| A | — | B |
| B | — | A |
| C | A B | — |
| D | C | C |
| E | D | — |
| F | E F | D F G |
| G | G | E |

Отмечаем в таблице и добавляем в очередь пары состояний, различаемых словом $\varepsilon$: все пары, один элемент которых — терминальное состояние, а второй — не терминальное состояние. Для данного автомата это пары

$(A, F), (B, F), (C, F), (D, F), (E, F), (A, G), (B, G), (C, G), (D, G), (E, G)$

Дальше итерируем процесс определения неэквивалентных состояний, пока очередь не оказывается пуста.

$(A, F)$ не дает нам новых неэквивалентных пар. Для $(B, F)$ находится 2 пары: $(A, D), (A, G)$. Первая пара не отмечена в таблице — отмечаем и добавляем в очередь. Вторая пара уже отмечена в таблице, значит, ничего делать не надо. Переходим к следующей паре из очереди. Итерируем дальше, пока очередь не опустошится.

Результирующая таблица (заполнен только треугольник, потому что остальное симметрично) и порядок добавления пар в очередь.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | | | | | | | |
| B | | | | | | | |
| C | ✓ | ✓ | | | | | |
| D | ✓ | ✓ | ✓ | | | | |
| E | ✓ | ✓ | ✓ | ✓ | | | |
| F | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| G | ✓ | ✓ | ✓ | ✓ | ✓ | | |

Очередь:
$(A, F), (B, F), (C, F), (D, F), (E, F), (A, G), (B, G), (C, G), (D, G), (E, G),$
$(B, D), (A, D), (A, E), (B, E), (C, E), (C, D), (D, E), (A, C), (B, C))$

В таблице выделились классы эквивалентных вершин: $\{A, B\}, \{C\}, \{D\}, \{E\}, \{F, G\}$. Остается только нарисовать результирующий автомат с вершинами-классами. Переходы добавляются тогда, когда из какого-нибудь состояния первого класса есть переход в какое-нибудь состояние второго класса. Минимизированный автомат: