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).

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 ε-переходов

(c) Минимальный полный детерминированный конечный автомат

Заметим, что из регулярного выражения можно выкинуть среднюю скобку. И будет тот же самый язык, так как:
Если слово S принадлежало

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

то его очевидно можно сконструировать с помощью

$$(a \mid b)^+(a \mid b)^+$$

Нужно просто среднюю часть $(aa \mid bb \mid abab \mid baba)^*$ 'занести' в $(a \mid b)^+$ (мы же там можем произвольное количество букв a и b использовать)
И обратно, если слово S принадлежало

$$(a \mid b)^+(a \mid b)^+$$

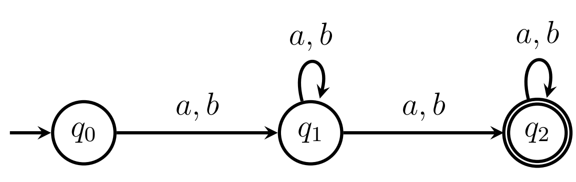
то его можно составить и с помощью

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

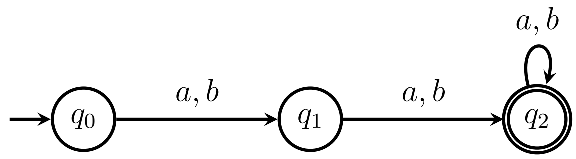
Так как средняя часть со звёздочкой, то можно просто повторить её 0 раз.

(a) (b) Недетерминированный конечный автомат без ε-переходов для

$$(a \mid b)^+(a \mid b)^+$$

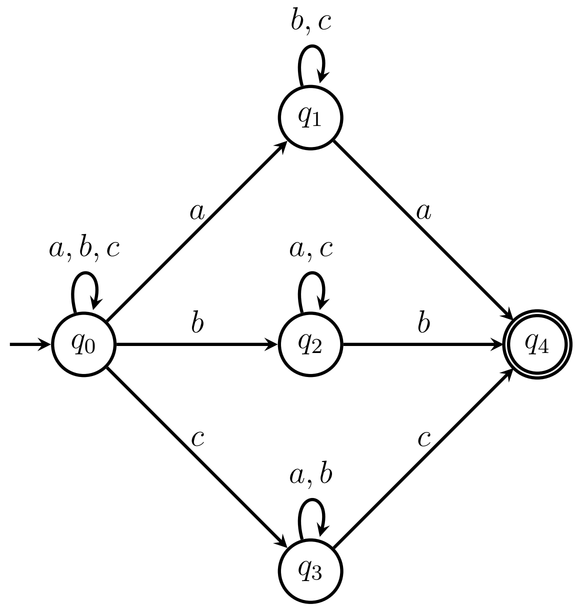


(c) Минимальный полный детерминированный конечный автомат



Понятно, что меньше 3 вершин быть не может, так как у нас есть две скобочки со знаком +, а значит во всех словах есть как минимум 2 буквы, а значит нужно 3 вершины.

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



Вершина q_0 даёт нам скобочку $(a|b|c)^*$
А дальше у нас есть 3 пути до q_4, перебираем каждый из них: $a(b|c)^*a$, $b(a|c)^*b$, $c(a|b)^*c$
Итого:

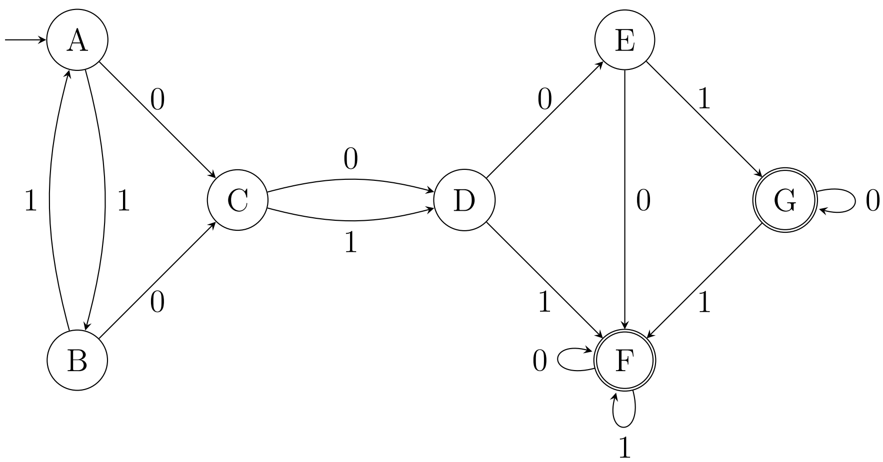
$$(a|b|c)^*((a(b|c)^*a) \mid (b(a|c)^*b) \mid (c(a|b)^*c))$$

4. Определить, является ли автоматным язык $\{\omega\omega^r \mid \omega \in \{0,1\}^*\}$. Если является — построить автомат, иначе — доказать.
Пусть он автоматный. Тогда выполняется условие леммы о накачке. Возьмём $w = 1^n001^n$, где n - из леммы
w лежит в языке, так как $\omega = 1^n0$
Так как $|w| > n$, то существует разбиение $xyz = w$, причём y не пустой. Так как по условию леммы $|xy| < n$, то y точно содержится в первых n единичках. Поэтому мы можем накачать w повторив y $k >= 2$ раза. Тогда будем получать слова вида $1^{n+c}001^n$
Такие слова точно не лежат в языке, так как у нас всего два нуля. А значит омега должна содержать ровно один ноль. А значит относительно этих нулей слово должно быть симметричным, но $c + n > n$, так как y не пуст
Противоречие с тем, что язык регулярный.

5. Определить, является ли автоматным язык $\{uaav \mid u, v \in \{a,b\}^*, |u|_b \geq |v|_a\}$. Если является — построить автомат, иначе — доказать.
   Пусть он автоматный. Тогда выполняется условие леммы о накачке. Возьмём $w = b^n aa(ba)^n$, где n - из леммы
   w лежит в языке, так как $u = b^n,\ v = (ba)^n$, $w = uaav$ и $|u|_b = |v|_a = n$
   Так как $|w| > n$, то существует разбиение $xyz = w$, причём y не пустой. Так как по условию леммы $|xy| < n$, то y точно содержит только буквы b и причём точно содержит хотя бы одну.
   То есть если мы возьмём xy^0z, то количество букв b в u строго уменьшиться. При этом разбиение на u и v однозначное, так как в нашем слове есть всего одно место, где встречаются две буквы a подряд : $bbb...b\ aa\ bababa...ba$
   А то есть теперь $|u|_b < n = |v|_a$.
   Полученное слово по условию леммы должно лежать в языке, но он не лежит. Противоречие с тем, что язык регулярный.

Пример применения алгоритма минимизации

Минимизируем данный автомат:



Автомат полный, в нем нет недостижимых вершин — продолжаем.

Строим обратное δ отображение.

δ^{-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 ε: 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: