

**Кнут, Ахо и все-все-
все**

**Какие способы поиска всех вхождений
паттерна длины M в тексте длины N
вы можете назвать?**

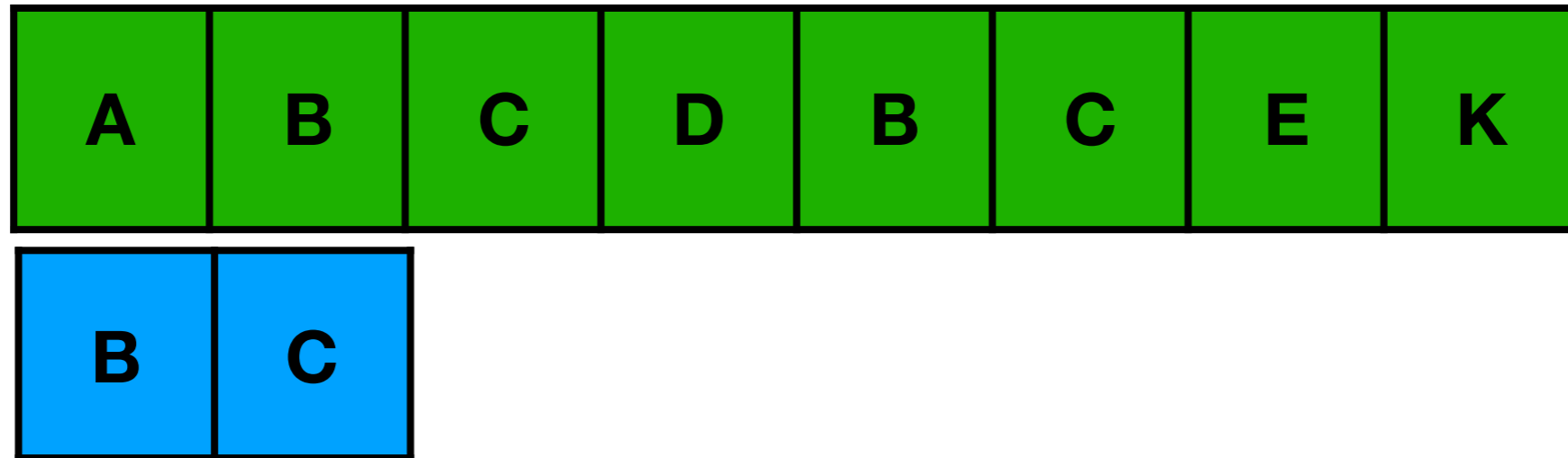
Наивный поиск

Ищем подстроку в строке, начиная сравнения с каждой позици

Какое время работы в лучшем случае?

Наивный поиск

Ищем строку, начиная заново с каждой позиции



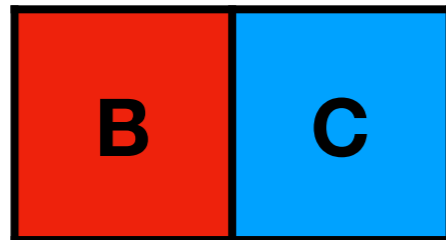
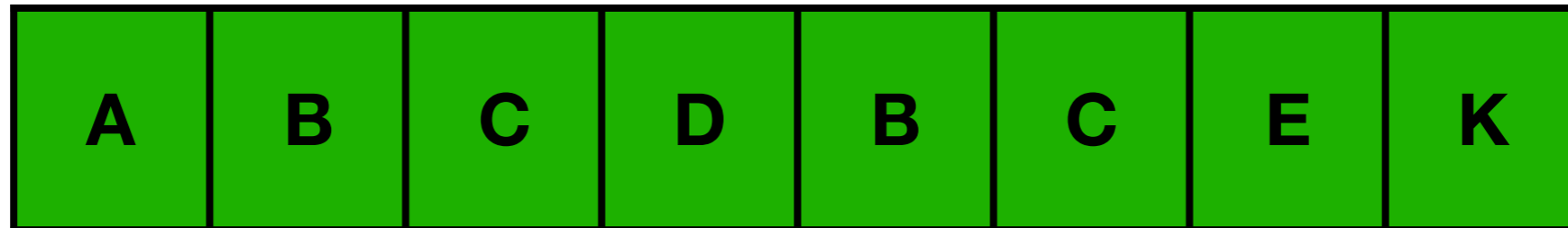
Какое время работы в лучшем случае?

Какое время работы в среднем случае?

Какое время работы в худшем случае?

Наивный поиск

Ищем строку, начиная заново с каждой позиции



Прекратили текущее сравнение

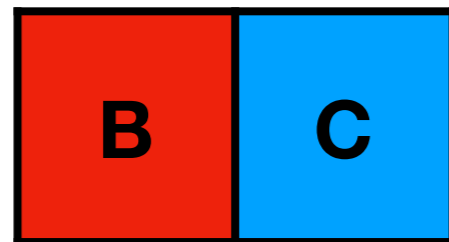
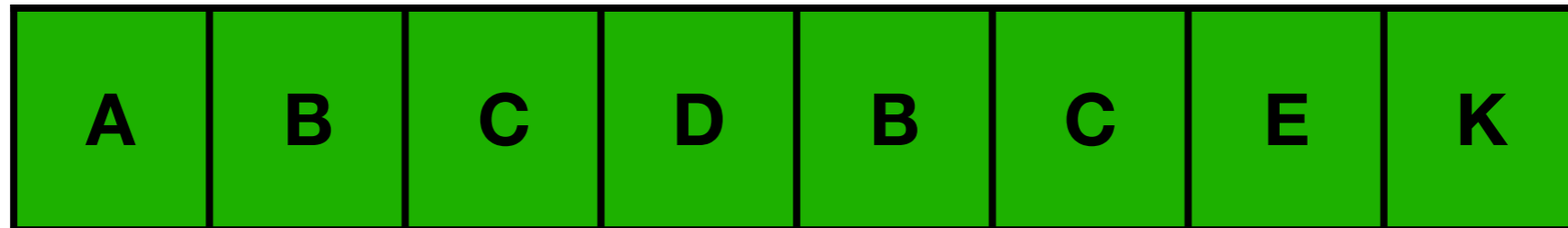
Какое время работы в лучшем случае?

Какое время работы в среднем случае?

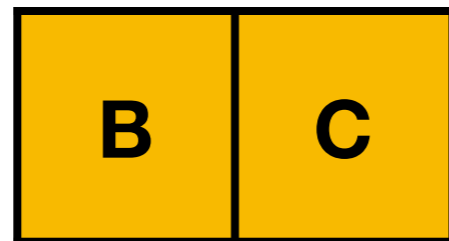
Какое время работы в худшем случае?

Наивный поиск

Ищем строку, начиная заново с каждой позиции



Прекратили текущее сравнение



Нашли паттерн

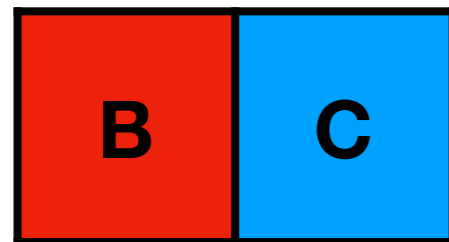
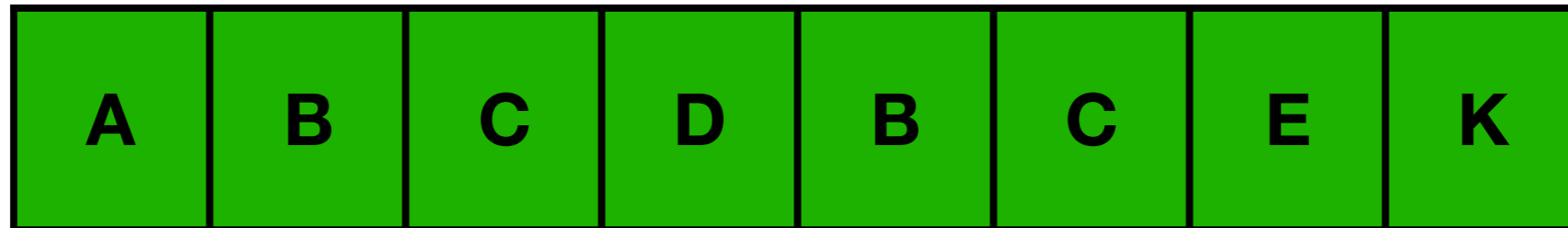
Какое время работы в лучшем случае?

Какое время работы в среднем случае?

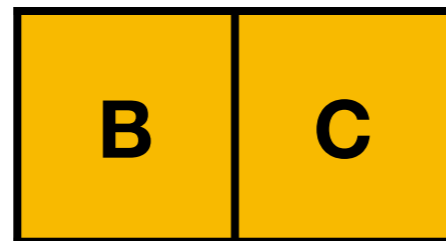
Какое время работы в худшем случае?

Наивный поиск

Ищем строку, начиная заново с каждой позиции



Прекратили текущее сравнение



Нашли паттерн

...

Какое время работы в лучшем случае?

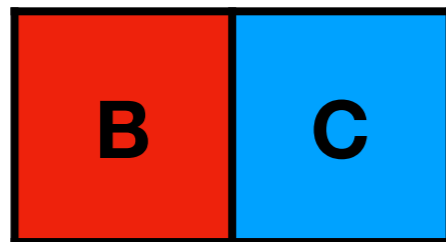
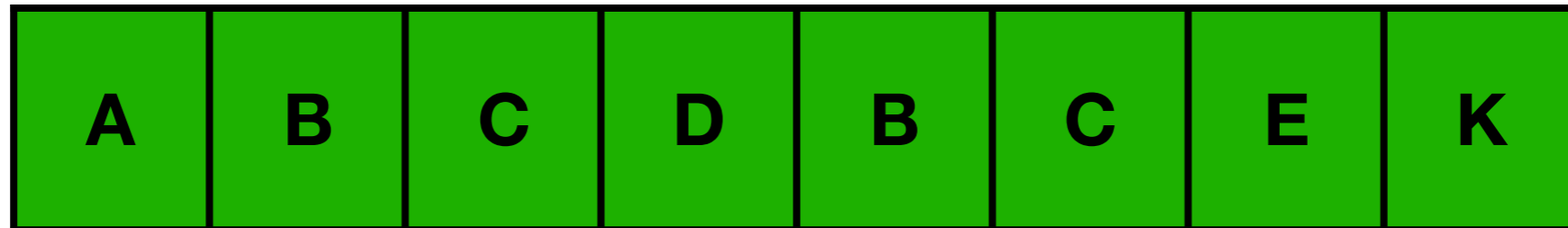
Прекратили текущее сравнение, конец строки

Какое время работы в среднем случае?

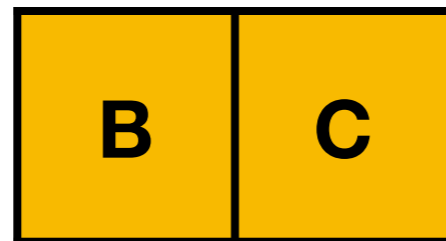
Какое время работы в худшем случае?

Наивный поиск

Ищем строку, начиная заново с каждой позиции

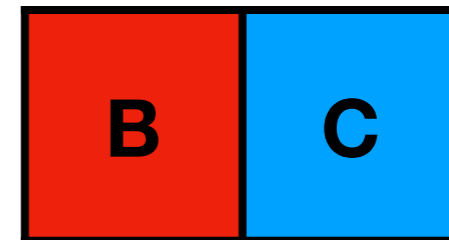


Прекратили текущее сравнение



Нашли паттерн

...



Какое время работы в лучшем случае?

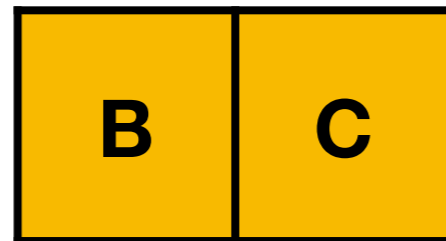
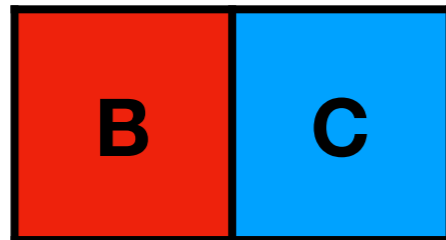
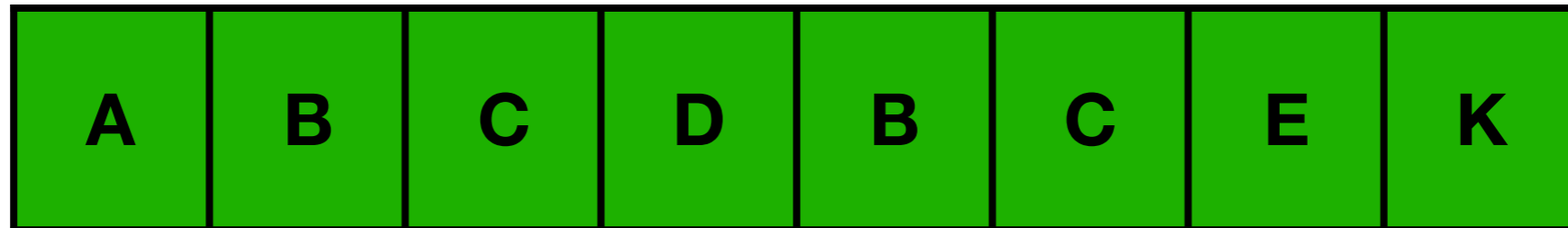
Какое время работы в среднем случае?

Какое время работы в худшем случае?

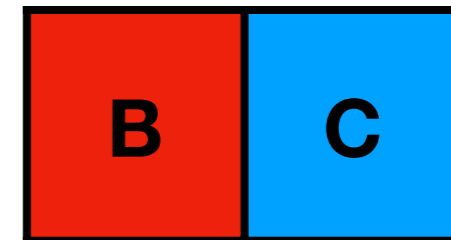
Прекратили текущее сравнение, конец строки

Наивный поиск

Ищем строку, начиная заново с каждой позиции



...



Какое время работы в лучшем случае?

$O(N)$

Какое время работы в среднем случае?

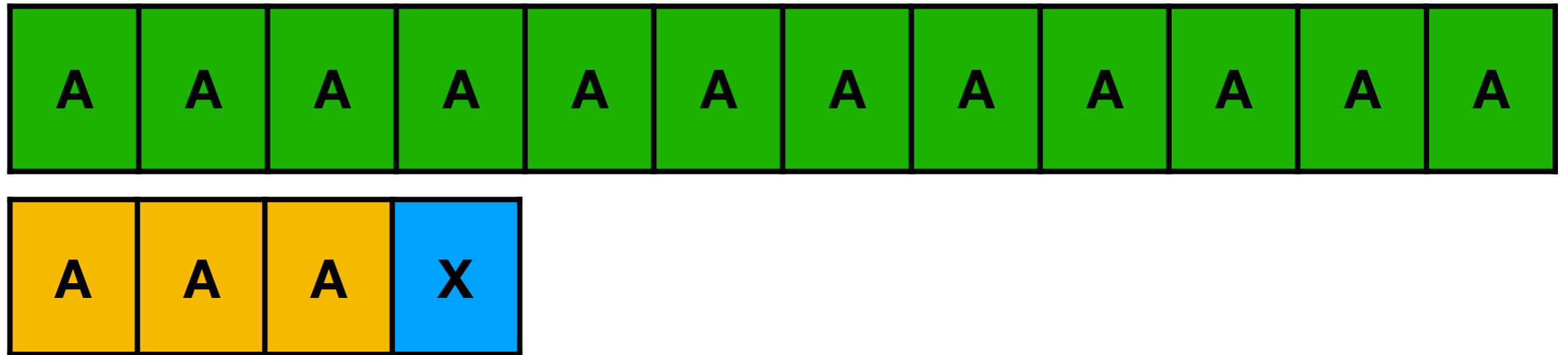
$O(N)$

Какое время работы в худшем случае?

$O(NM)$

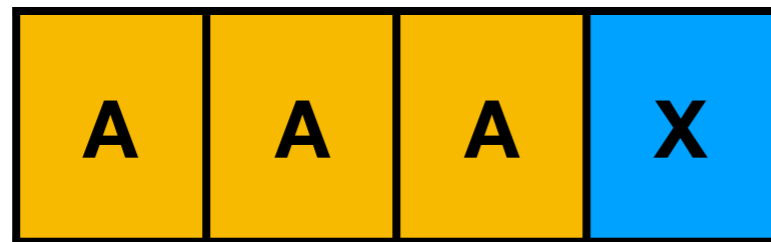
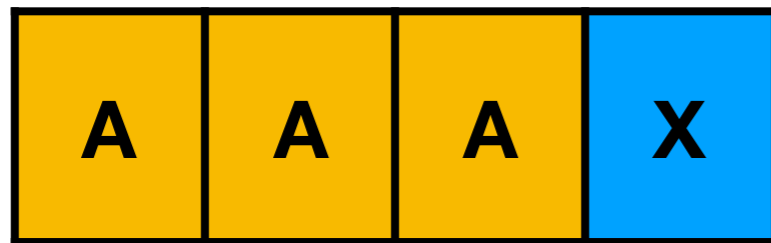
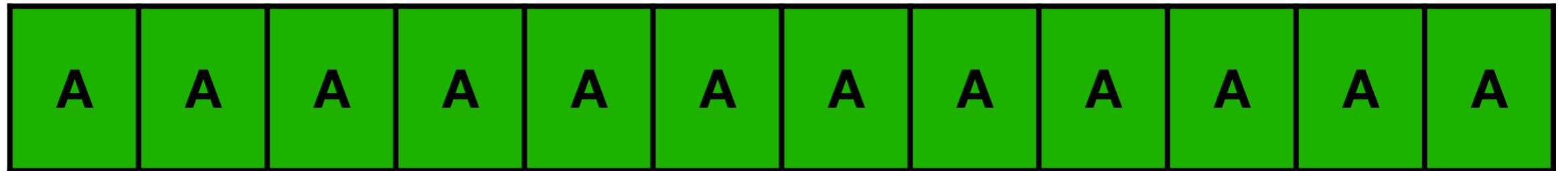
Наивный поиск

Худший случай



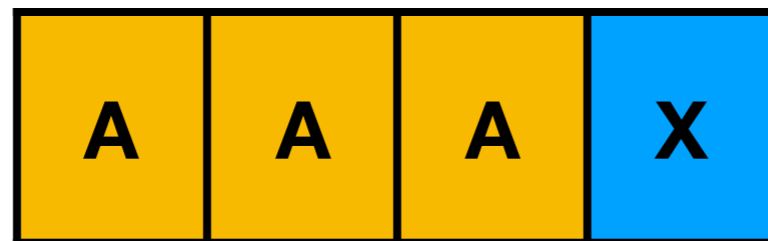
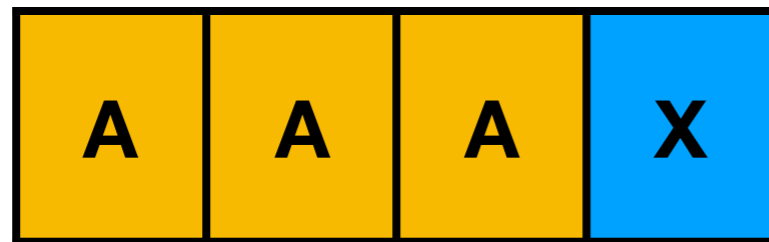
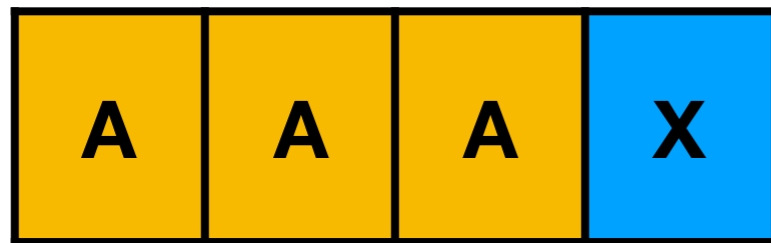
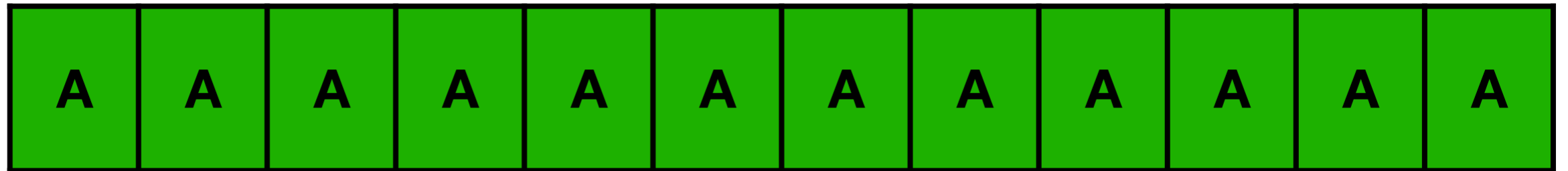
Наивный поиск

Худший случай



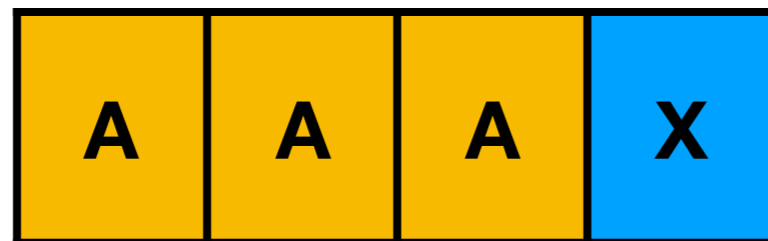
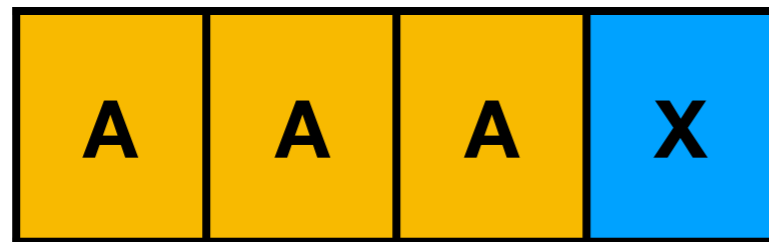
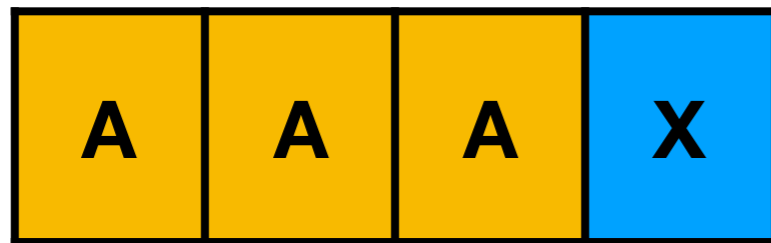
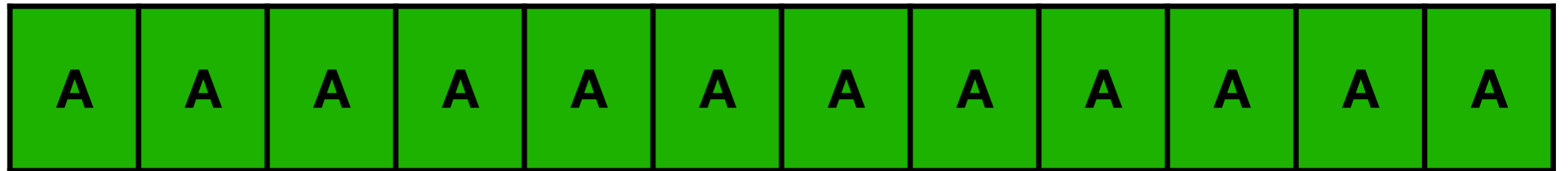
Наивный поиск

Худший случай



Наивный поиск

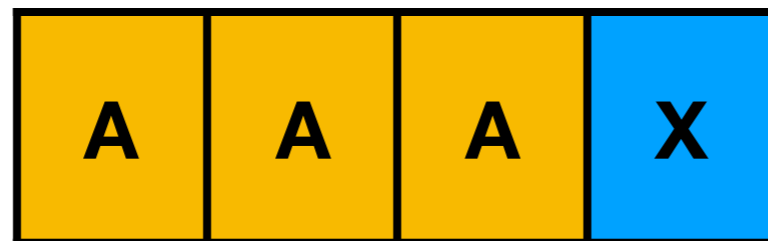
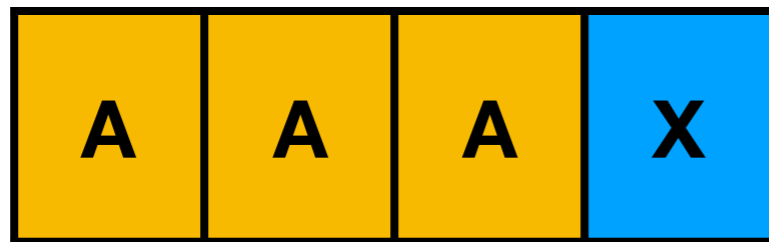
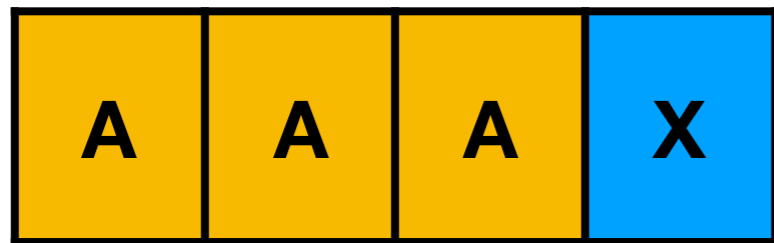
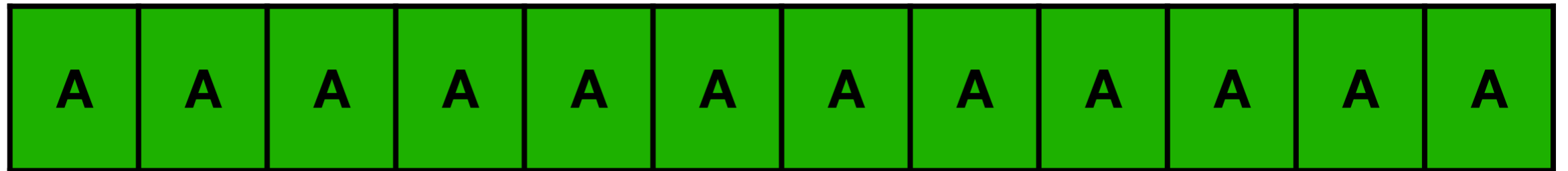
Худший случай



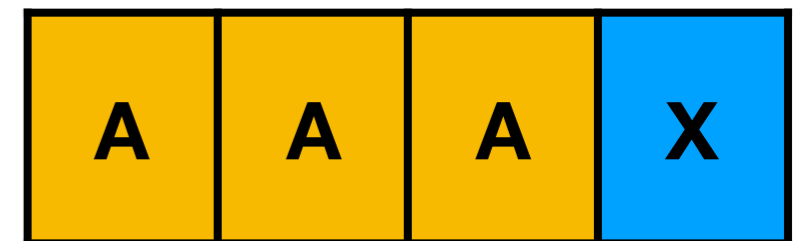
...

Наивный поиск

Худший случай



...



Алгоритм Кнута – Морриса – Пратта

Используем тот факт, что префикс-функция (длина наибольшего префикса подстроки $s[0:i]$, равного суффиксу подстроки $s[0:i]$) считается за $O(X)$ для всех позиций, где X - размер строки s

π

0	0	0	1	1	2	3	4	2
---	---	---	---	---	---	---	---	---

A	B	B	A	A	B	B	A	B
---	---	---	---	---	---	---	---	---

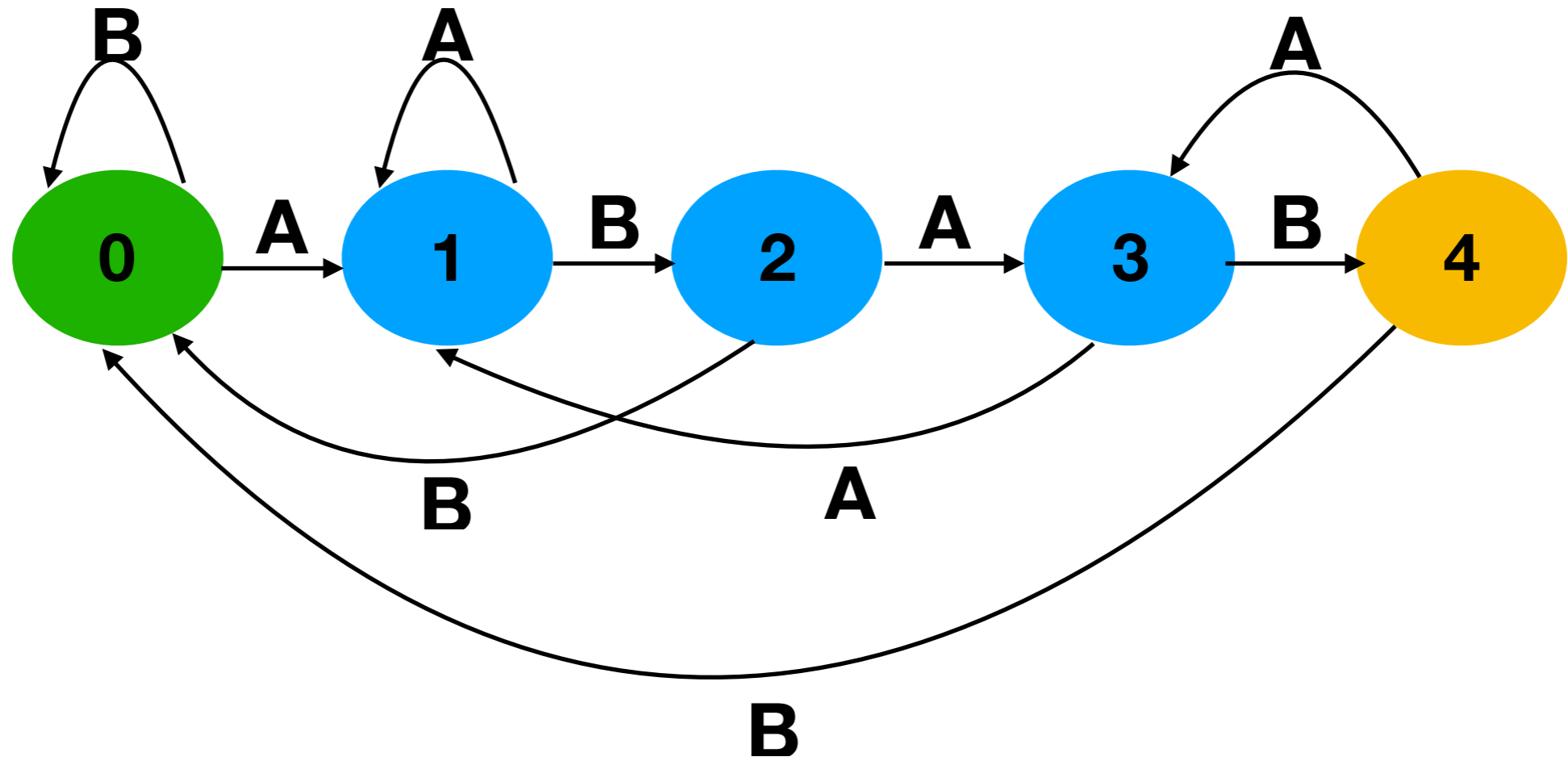
Алгоритм Кнута – Морриса – Пратта

Какое время работы в лучшем случае? **$O(N + M)$**

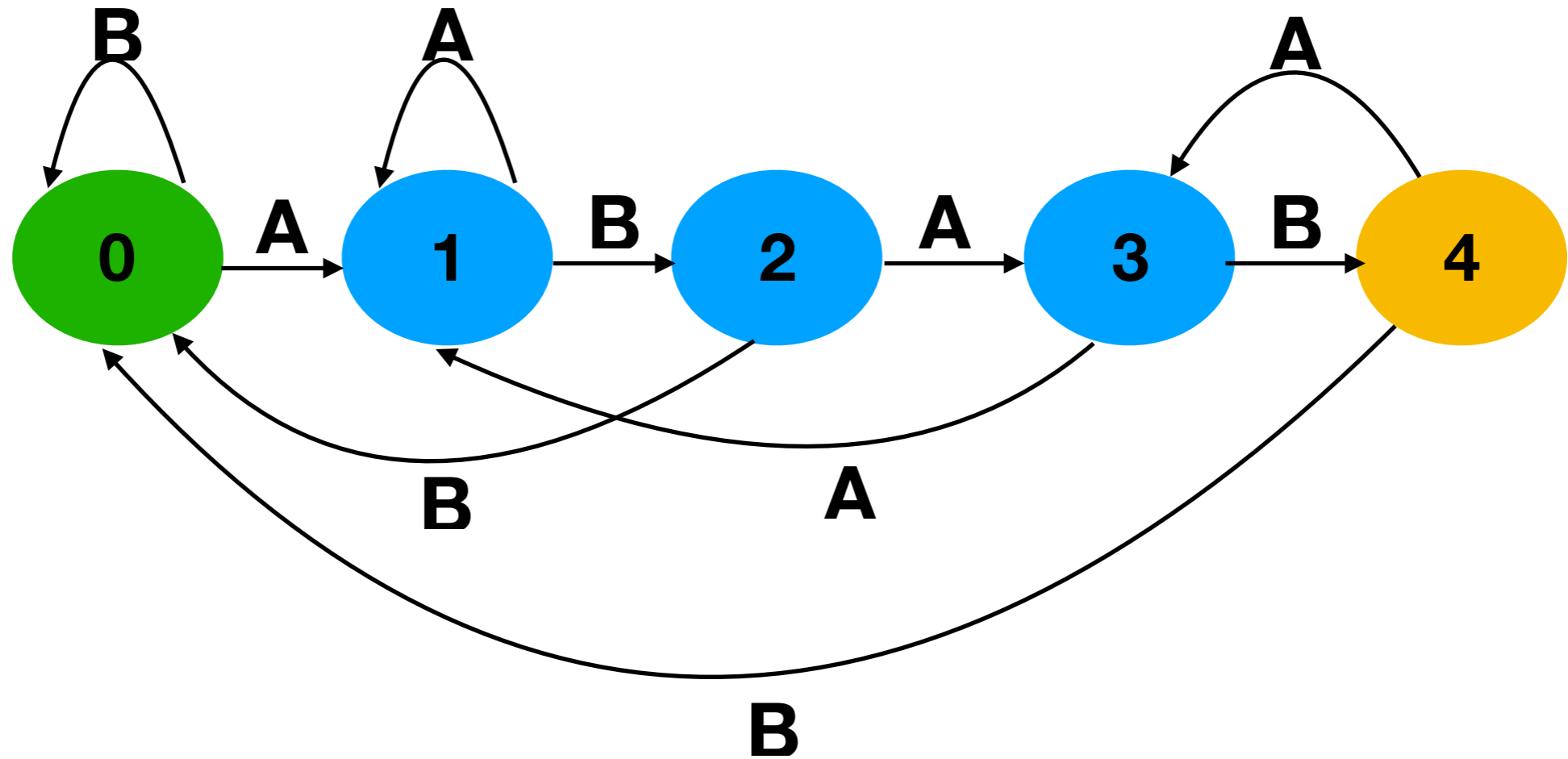
Какое время работы в среднем случае? **$O(N + M)$**

Какое время работы в худшем случае? **$O(N + M)$**

Конечный автомат



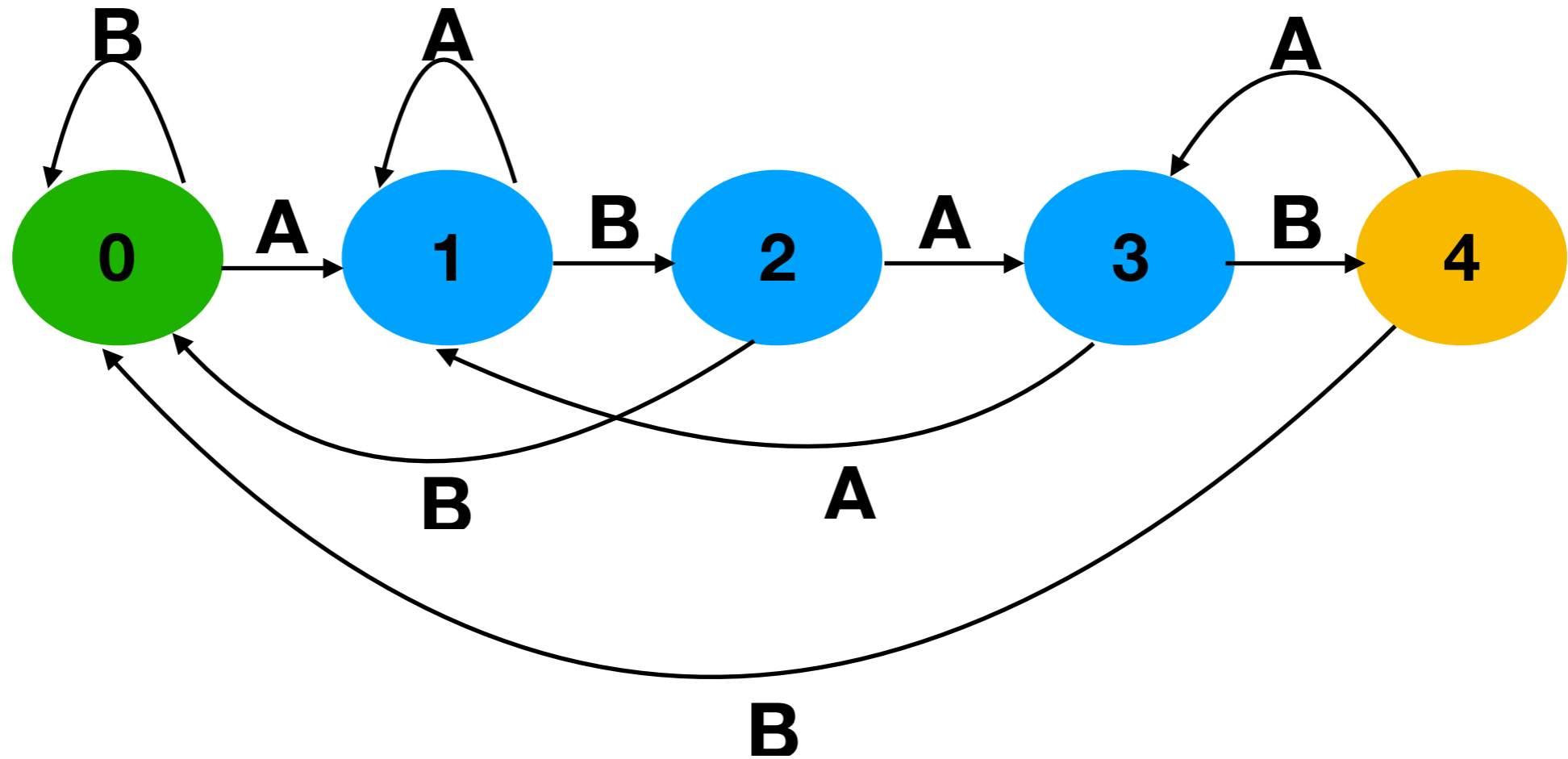
Конечный автомат



Состояние/ Символ	A	B
0	1	0
1	1	2
2	0	3
3	1	4
4	0	3

Сколько считаем таблицу переходов?

Конечный автомат



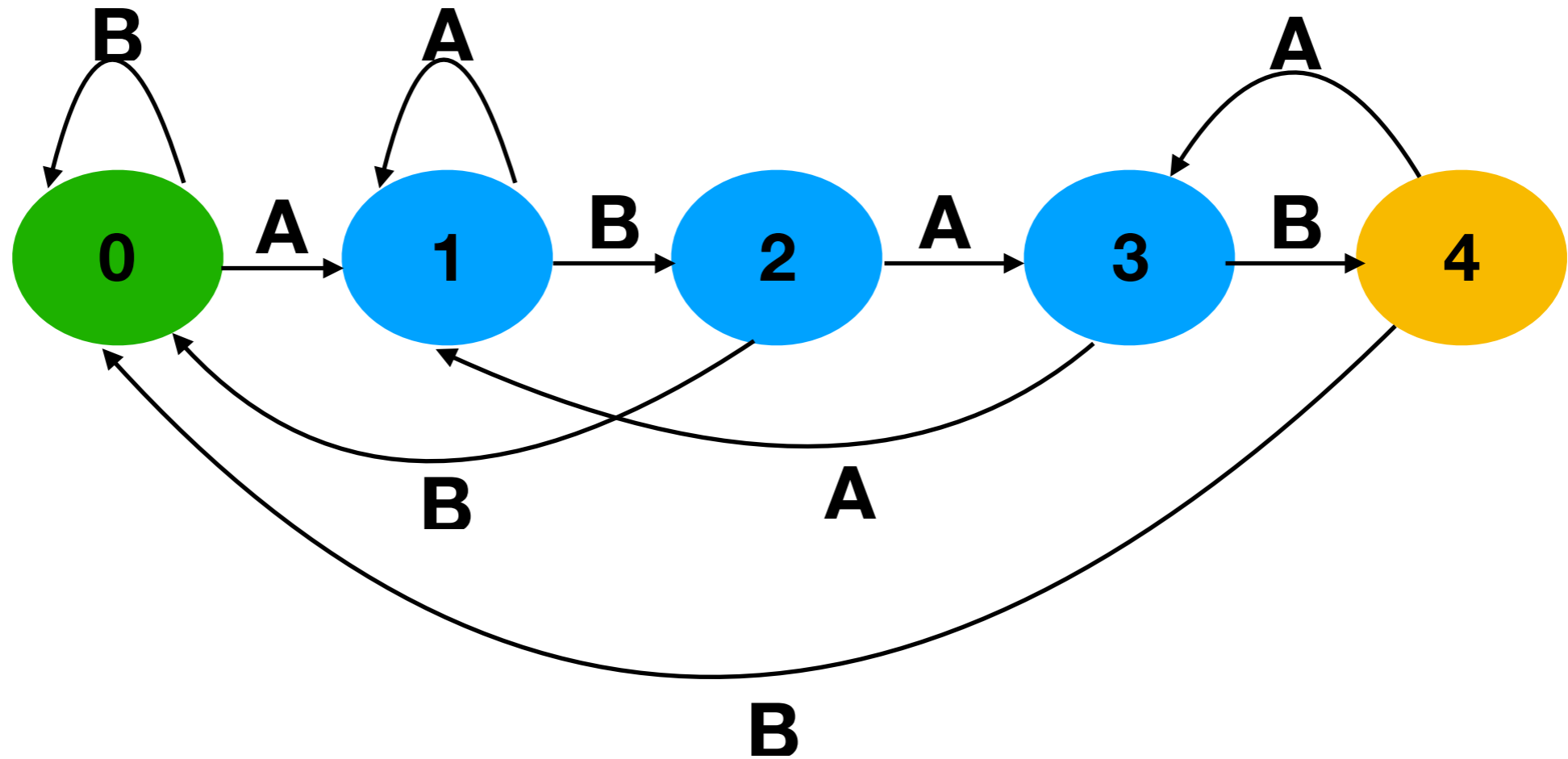
Состояние/ Символ	A	B
0	1	0
1	1	2
2	0	3
3	1	4
4	0	3

Сколько считаем таблицу переходов?

$$O(M^3 \cdot |\Sigma|) \quad ???$$

Пока оставим

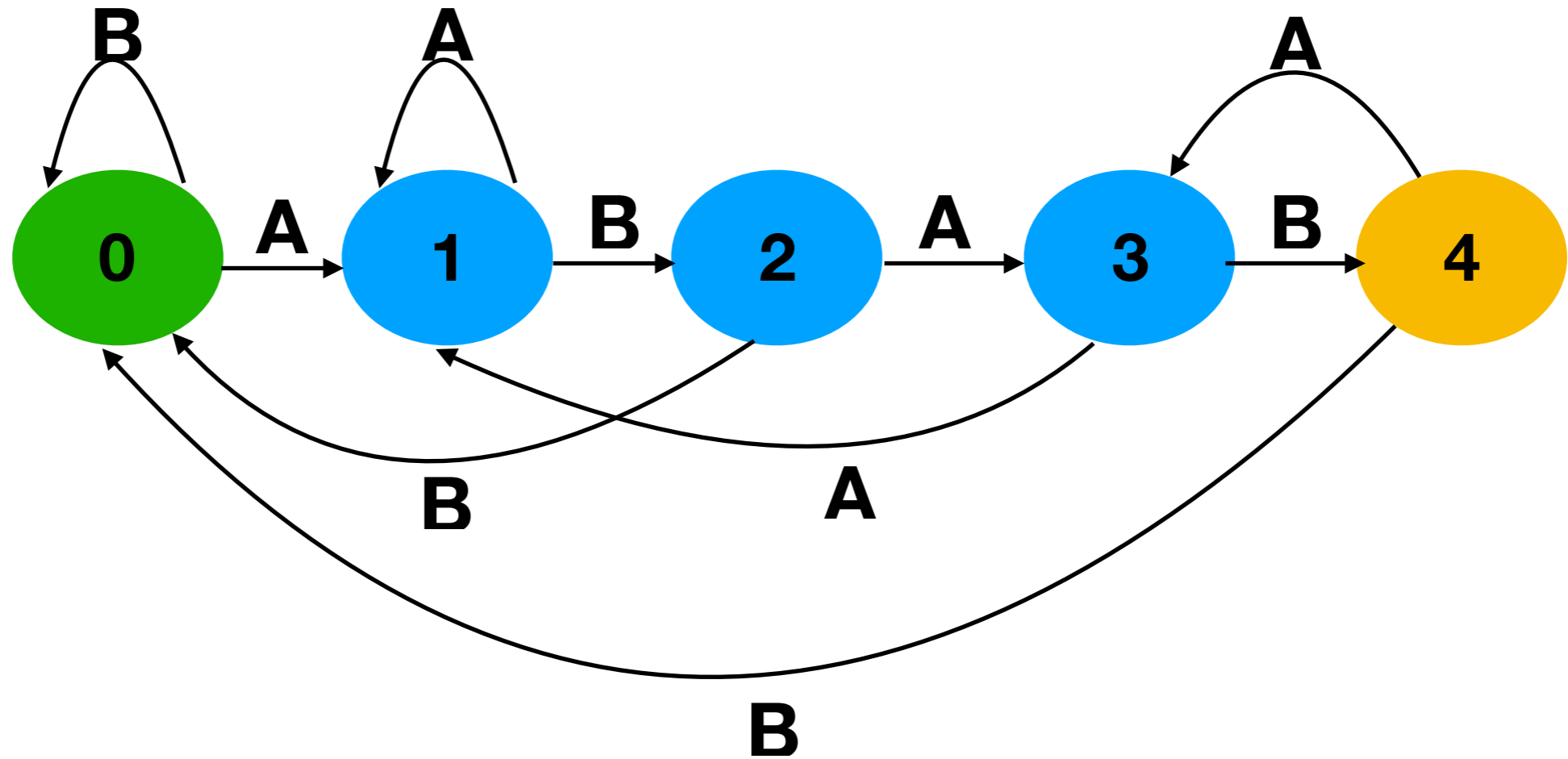
Конечный автомат



Состояние/ Символ	A	B
0	1	0
1	1	2
2	0	3
3	1	4
4	0	3

Сколько ищем в тексте длины N
(уже построенным автоматом)?

Конечный автомат



Состояние/ Символ	A	B
0	1	0
1	1	2
2	0	3
3	1	4
4	0	3

Сколько ищем в тексте длины N
(уже построенным автоматом)?

$$O(N)$$

Конечный автомат

Какое время работы в лучшем случае? $O(M^3 \cdot |\Sigma|) + O(N)$

Какое время работы в среднем случае? $O(M^3 \cdot |\Sigma|) + O(N)$

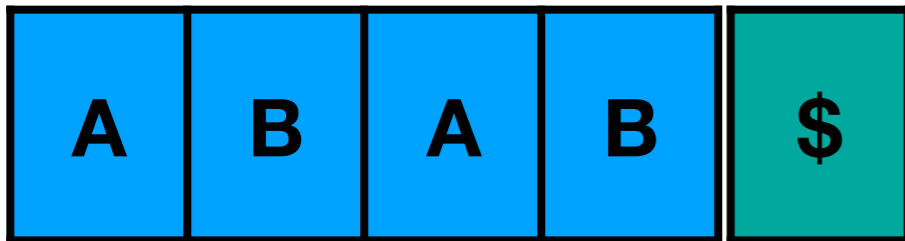
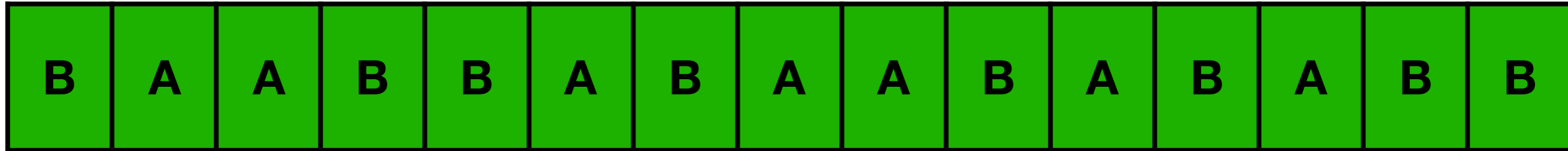
Какое время работы в худшем случае? $O(M^3 \cdot |\Sigma|) + O(N)$

Выглядит грустно. Пока размер текста не очень большой, толку нет. Проиграем даже наивному алгоритму

Кто фаворит?

Кнут-Моррис-Пратт

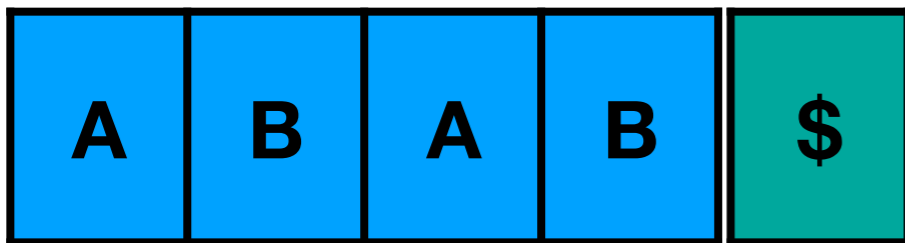
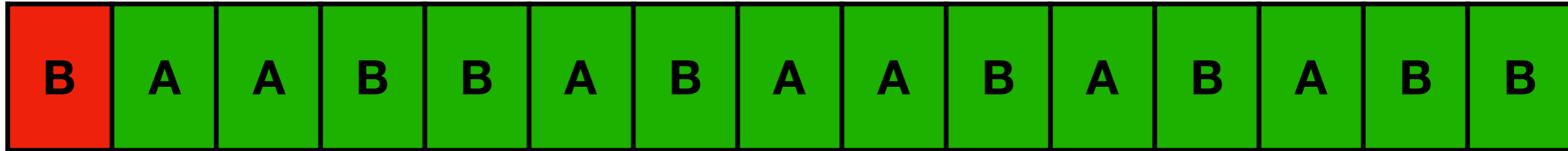
Давайте попробуем улучшить



ind	0	1	2	3
π	0	0	1	2

Кнут-Моррис-Пратт

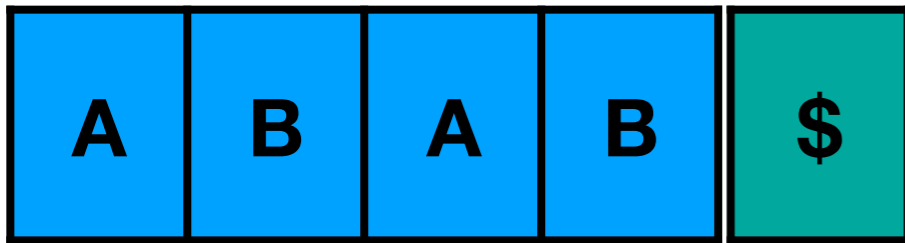
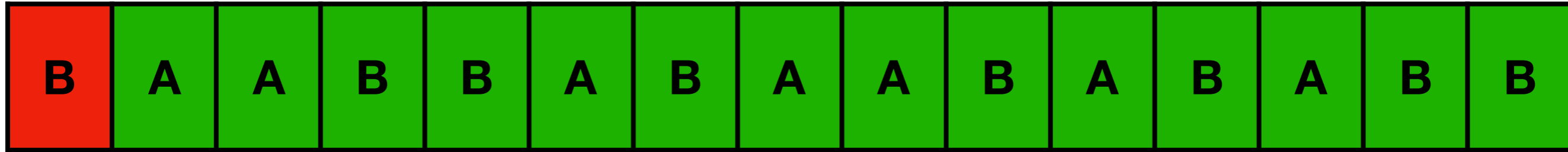
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A					
B					

Кнут-Моррис-Пратт

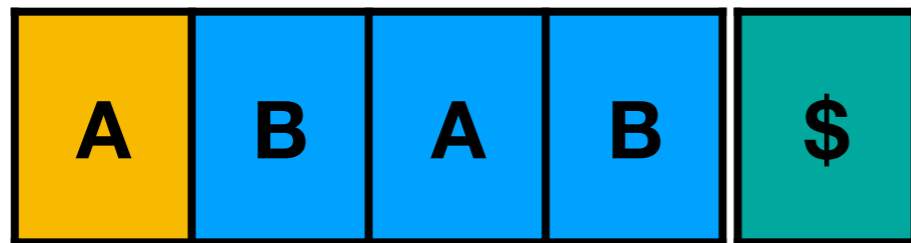
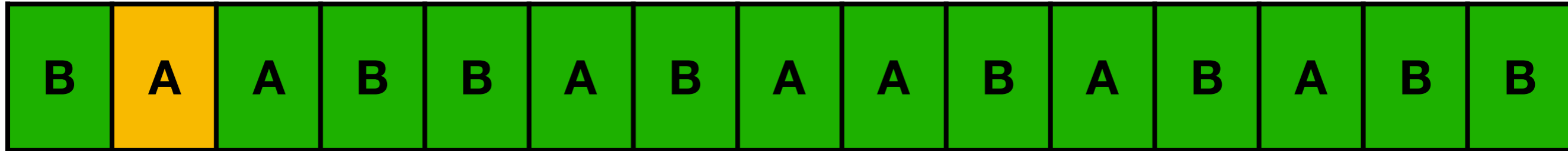
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A					
B	0				

Кнут-Моррис-Пратт

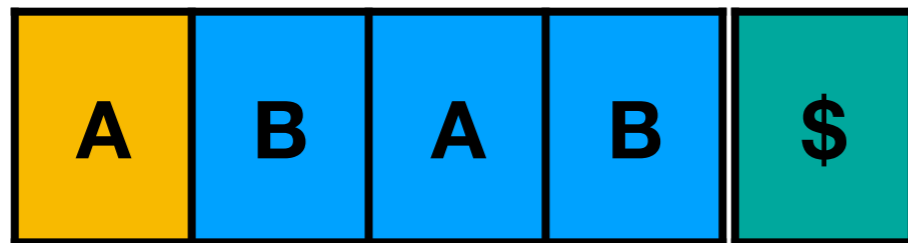
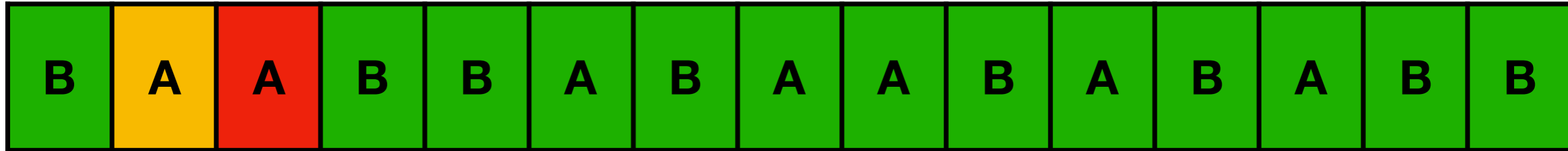
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1				
B	0				

Кнут-Моррис-Пратт

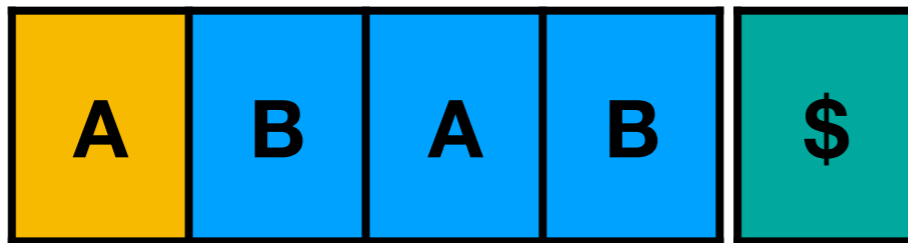
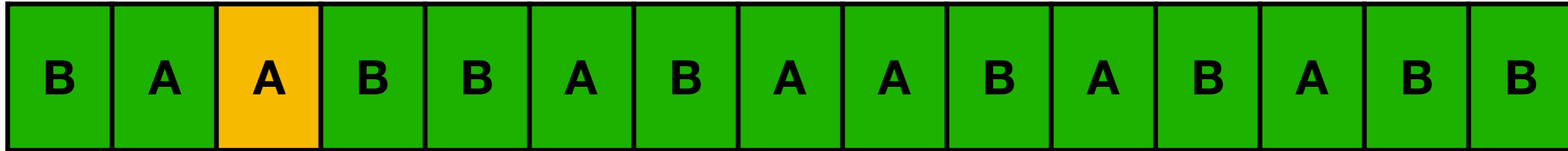
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1				
B	0				

Кнут-Моррис-Пратт

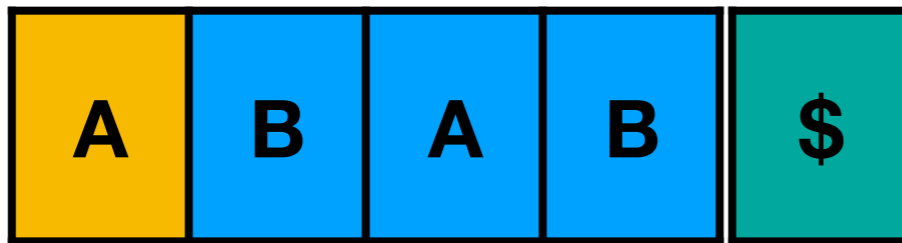
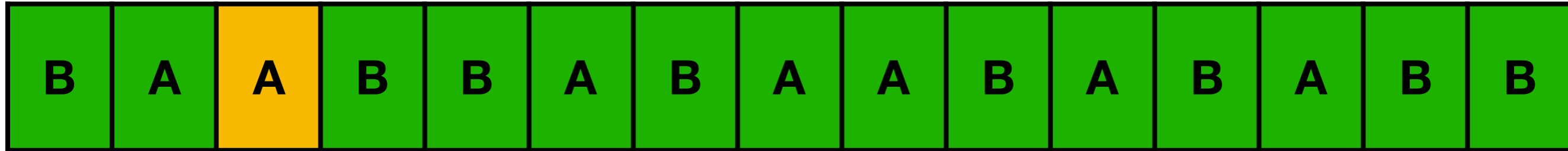
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1				
B	0				

Кнут-Моррис-Пратт

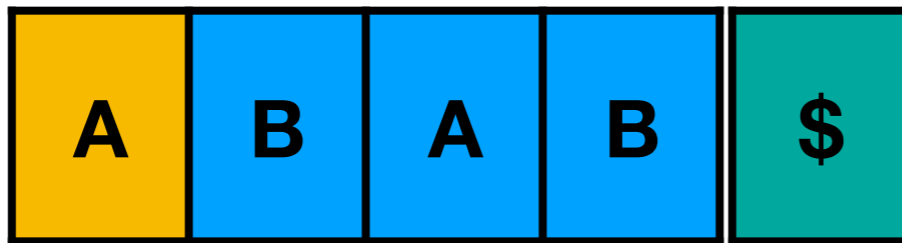
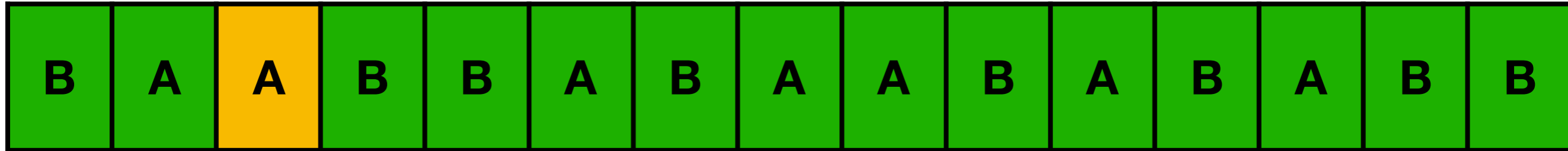
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0				

Кнут-Моррис-Пратт

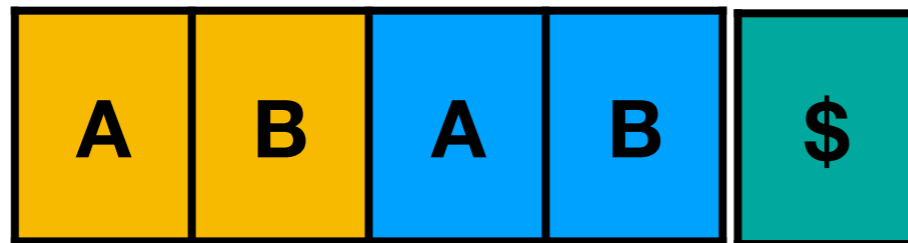
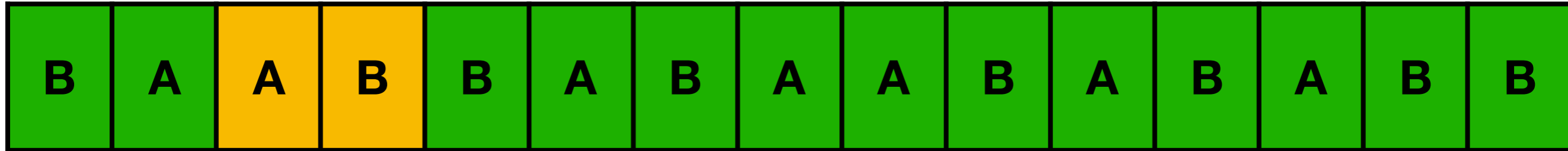
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0				

Кнут-Моррис-Пратт

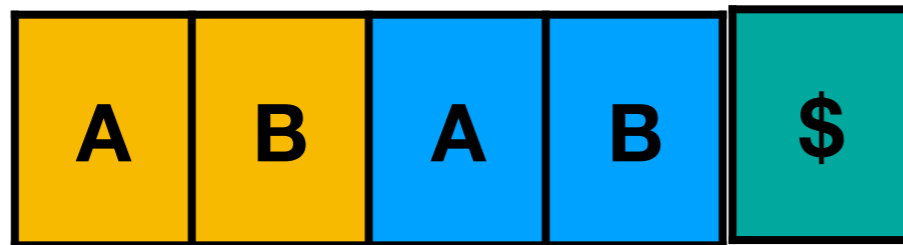
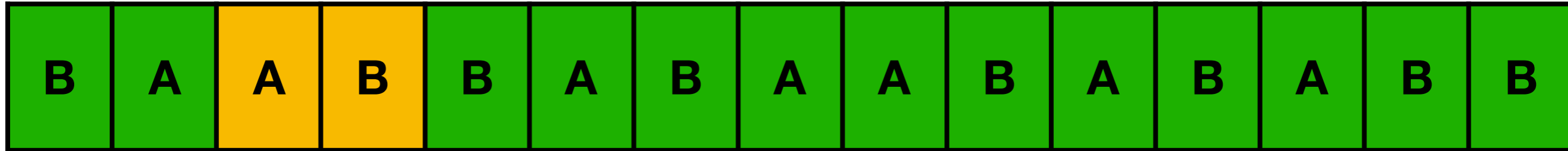
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0				

Кнут-Моррис-Пратт

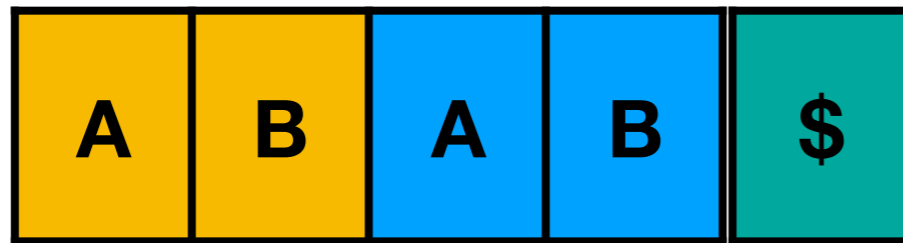
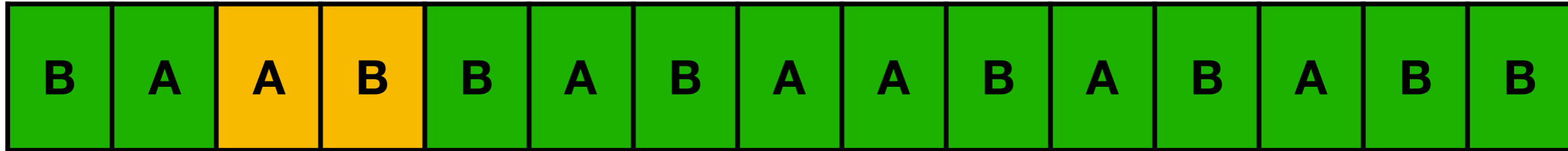
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2			

Кнут-Моррис-Пратт

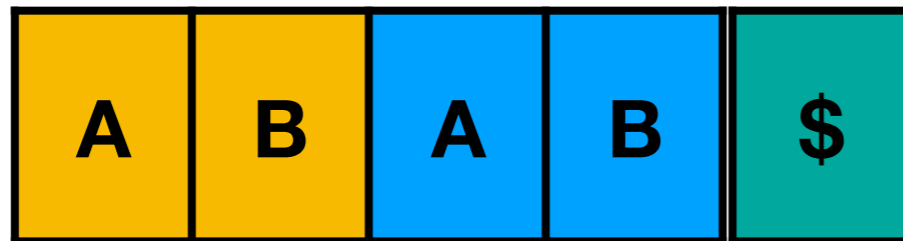
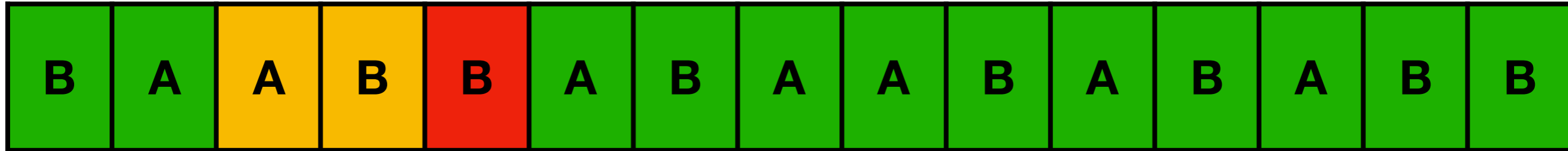
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2			

Кнут-Моррис-Пратт

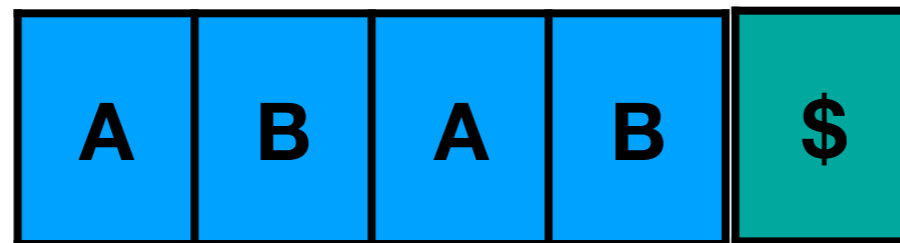
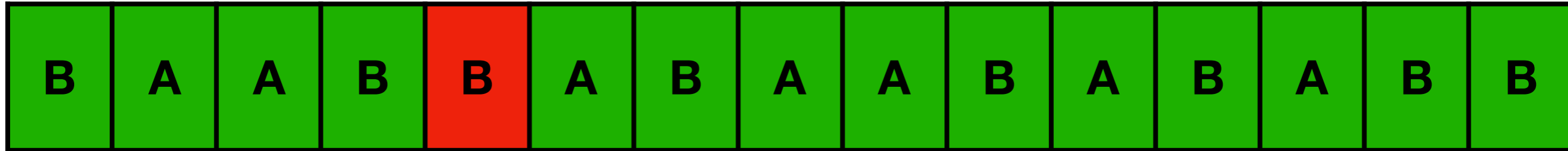
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2			

Кнут-Моррис-Пратт

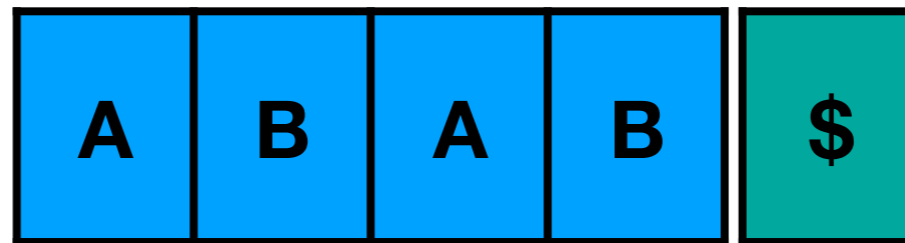
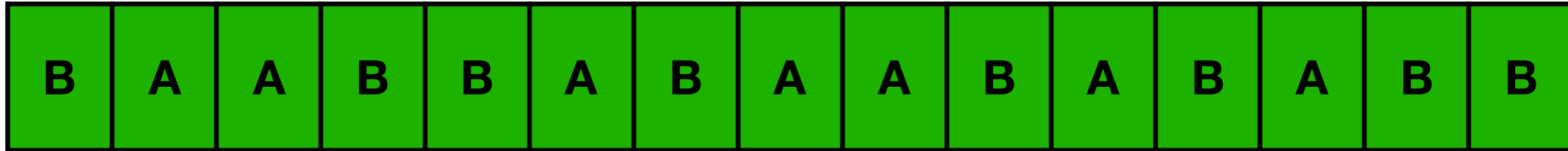
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2	0		

Кнут-Моррис-Пратт

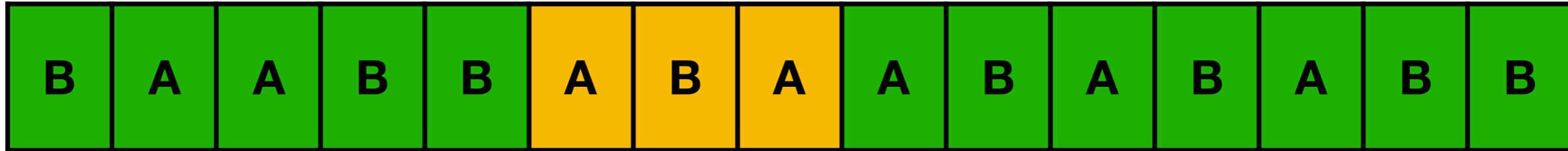
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2	0		

Кнут-Моррис-Пратт

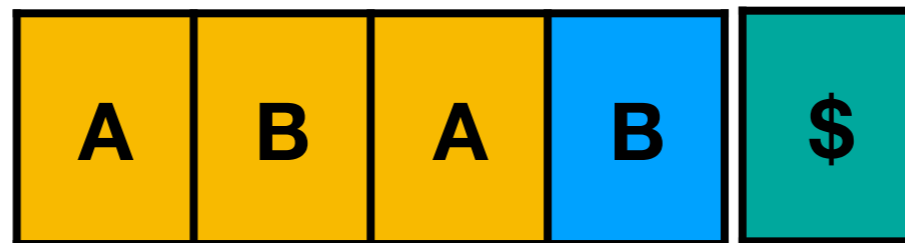
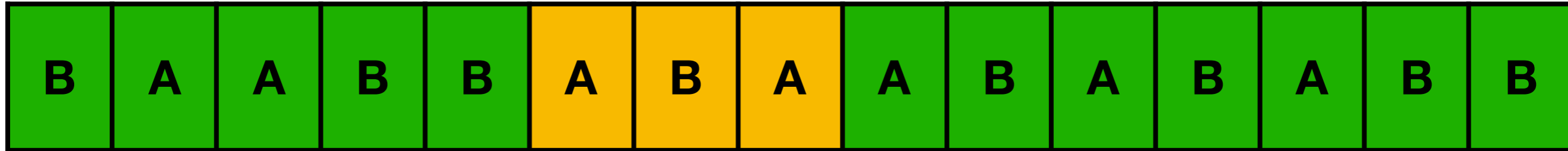
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1			
B	0	2	0		

Кнут-Моррис-Пратт

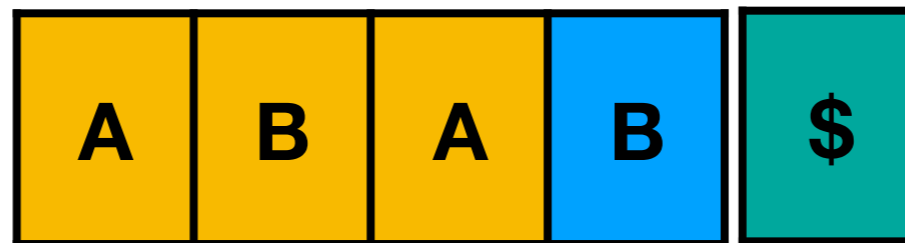
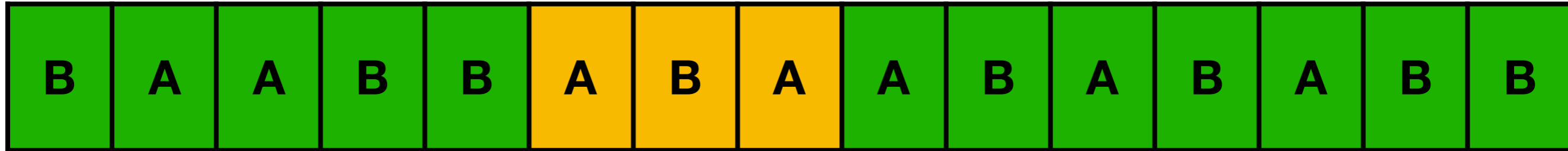
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3		
B	0	2	0		

Кнут-Моррис-Пратт

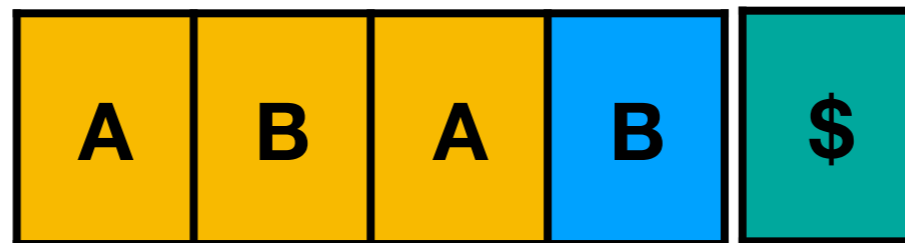
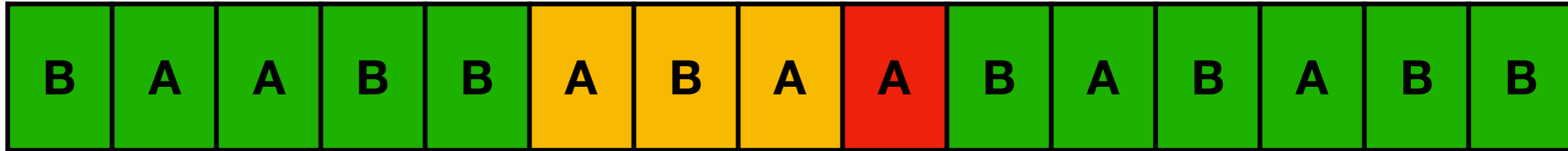
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3		
B	0	2	0		

Кнут-Моррис-Пратт

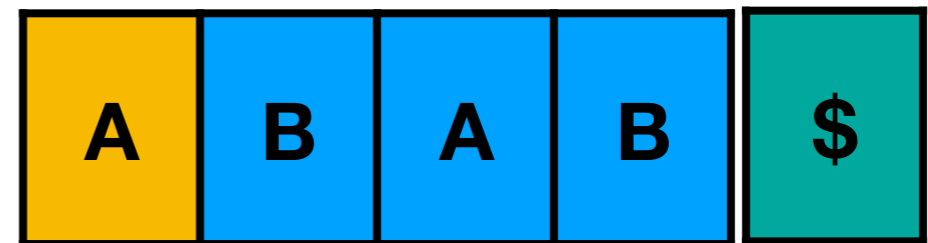
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3		
B	0	2	0		

Кнут-Моррис-Пратт

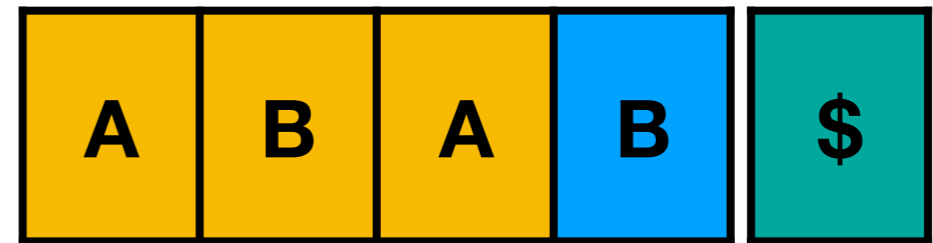
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	
B	0	2	0		

Кнут-Моррис-Пратт

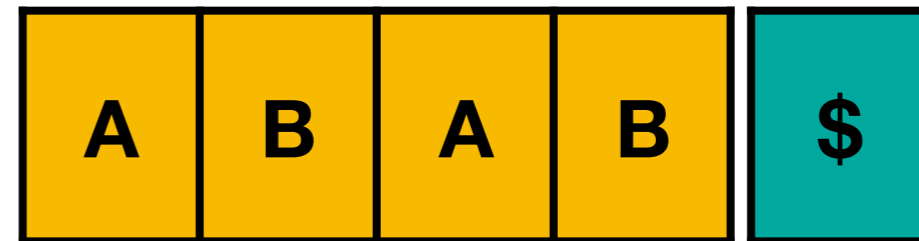
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	
B	0	2	0		

Кнут-Моррис-Пратт

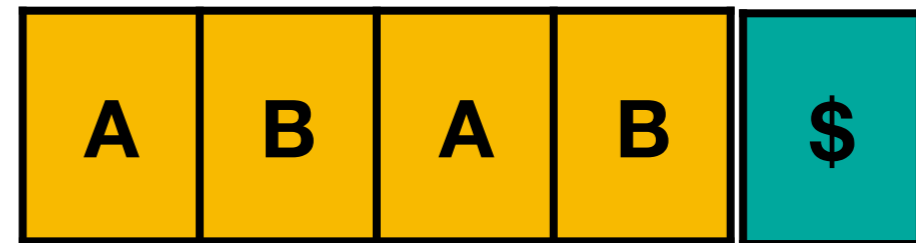
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	
B	0	2	0	4	

Кнут-Моррис-Пратт

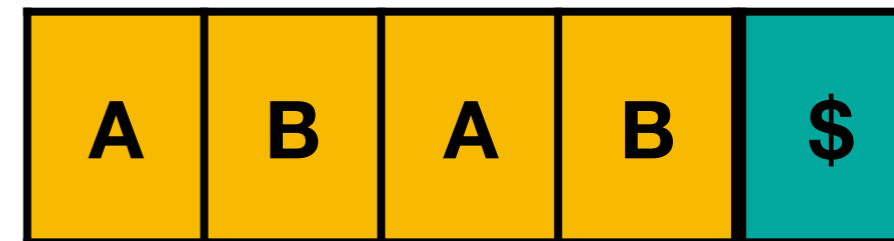
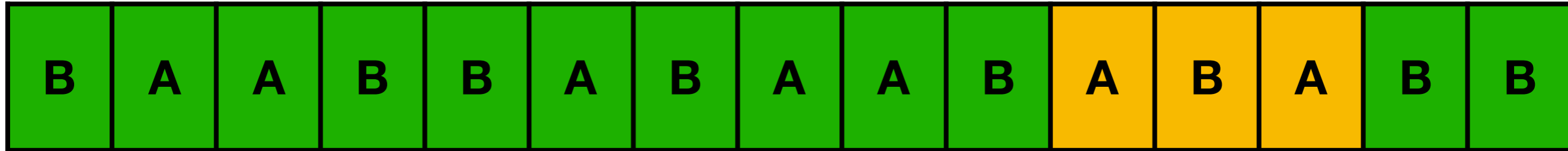
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	
B	0	2	0	4	

Кнут-Моррис-Пратт

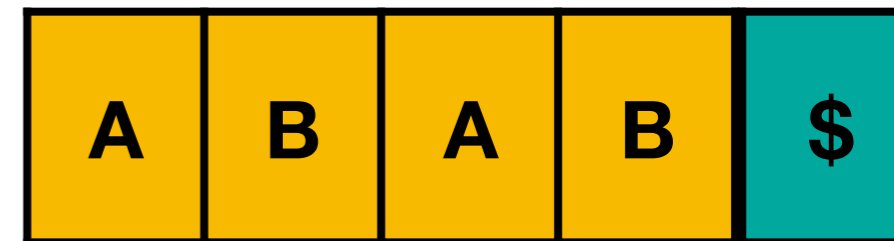
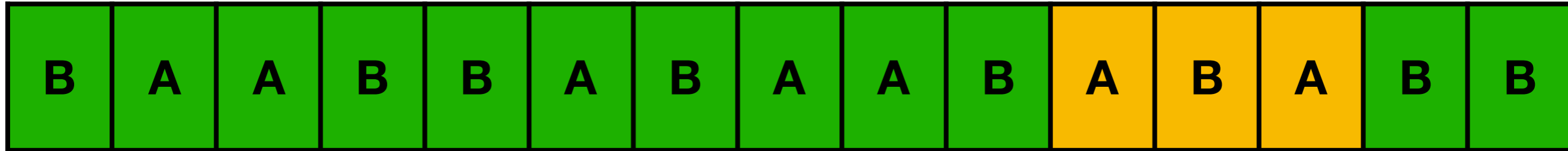
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

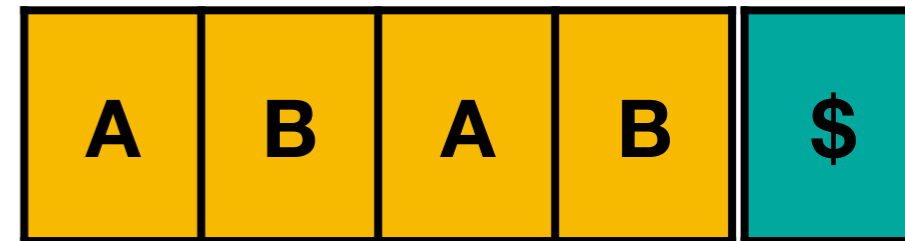
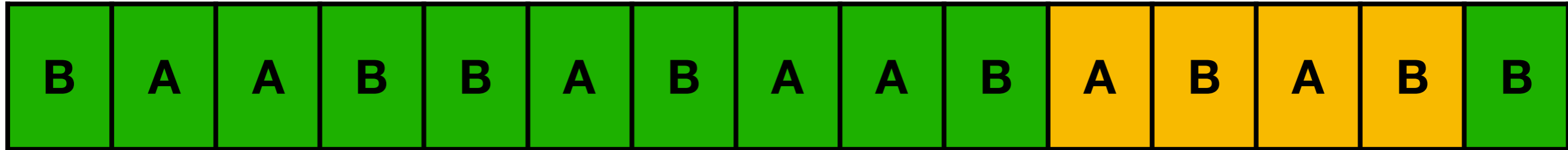
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

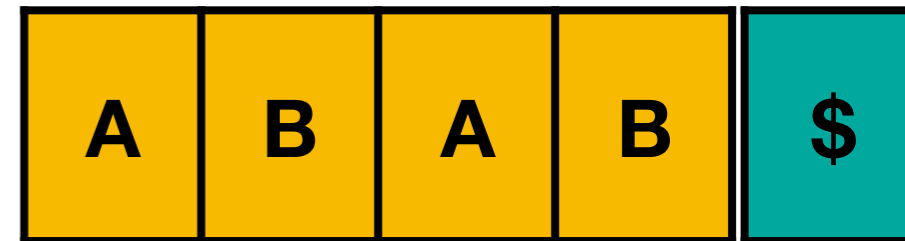
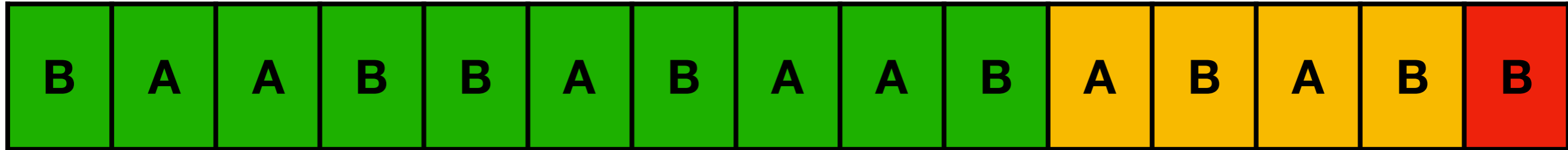
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

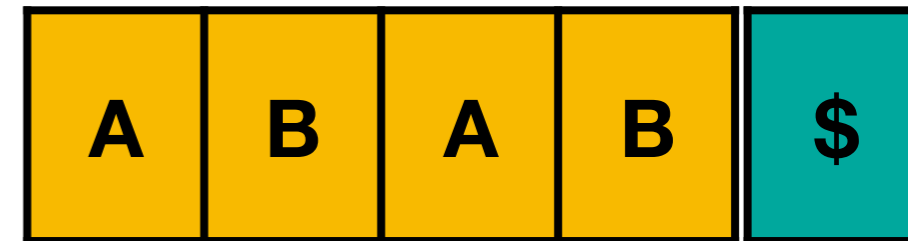
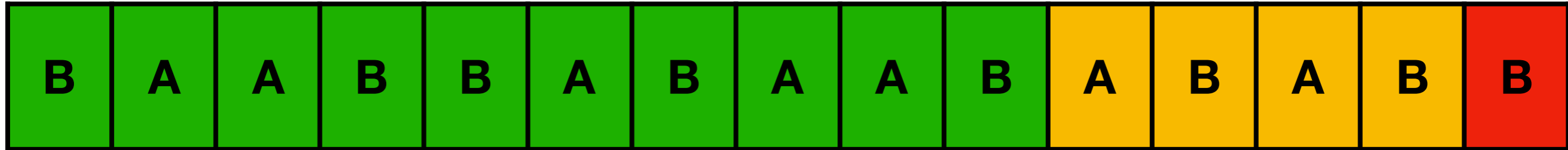
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

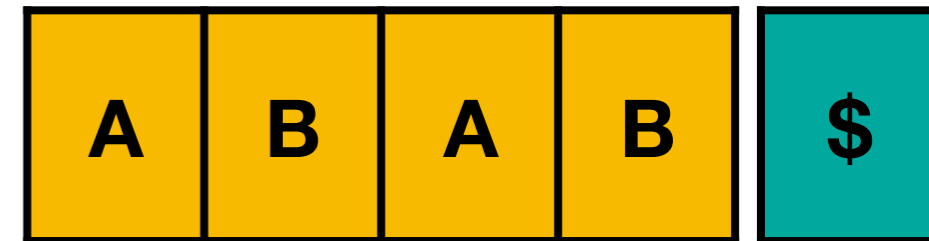
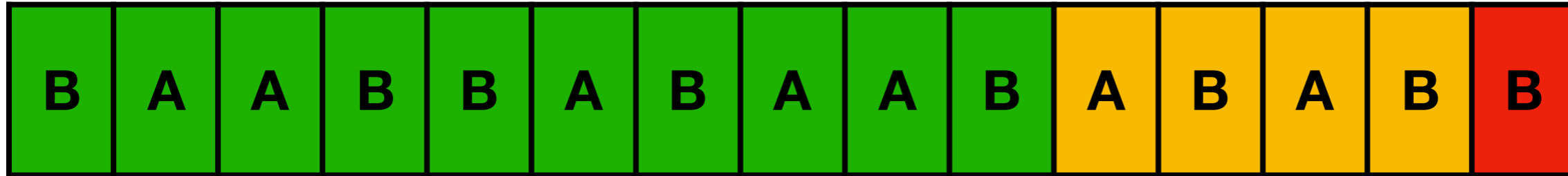
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

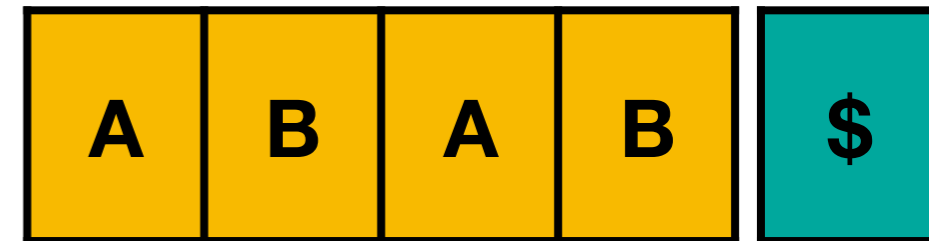
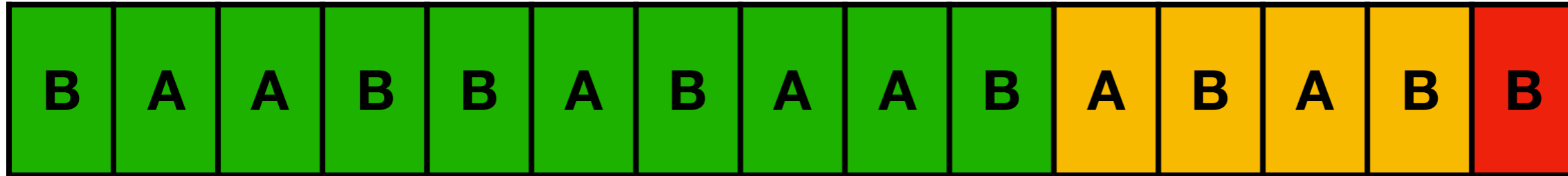
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	

Кнут-Моррис-Пратт

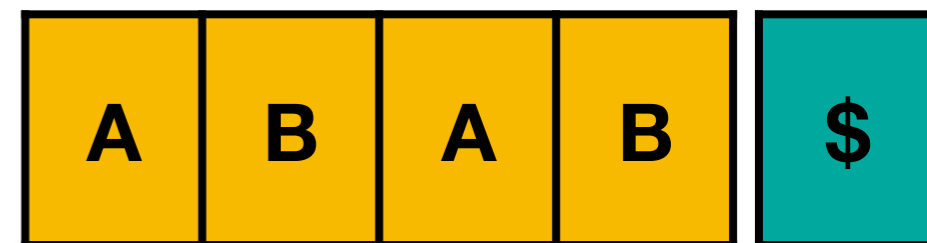
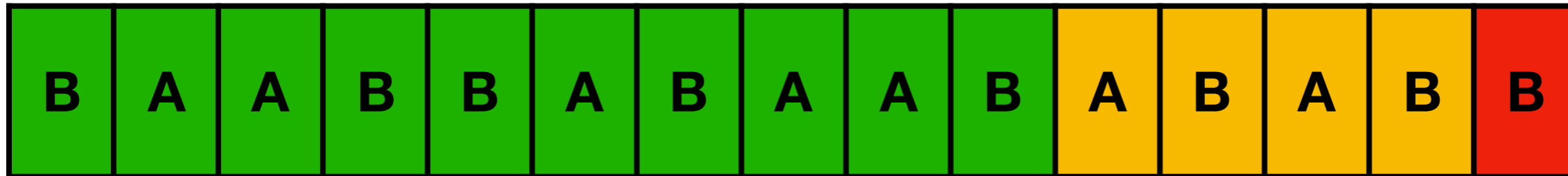
Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	0

Кнут-Моррис-Пратт

Давайте попробуем улучшить



ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	0

Кнут-Моррис-Пратт

В принципе - улучшили. Но не сильно (на константу). Но что мы посчитали в процессе?

ind	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	0

Кнут-Моррис-Пратт

В принципе - улучшили. Но не сильно (на константу). Но что мы посчитали в процессе?

СИМВОЛ/ СОСТОЯНИЕ	0	1	2	3	4
π	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	0

Кнут-Моррис-Пратт

В принципе - улучшили. Но не сильно (на константу). Но что мы посчитали в процессе?

СИМВОЛ/ СОСТОЯНИЕ	0	1	2	3	4
А	1	1	3	1	3
В	0	2	0	4	0

Таблицу переходов

Вычисление функции переходов

1. Вычислить префикс-функцию для паттерна
2. Вычислить функцию переходов для нулевого состояния

СИМВОЛ/ СОСТОЯНИ	0	1	2	3	4
<i>π</i>	0	0	1	2	
A	1				
B	0				

Вычисление функции переходов

3. $\sigma(state, symbol) = state + 1,$ *if* $s[state] = symbol$
 $\sigma(state, symbol) = \sigma(\pi[state - 1], symbol),$ *otherwise*

СИМВОЛ/ СОСТОЯНИ	0	1	2	3	4
π	0	0	1	2	
A	1				
B	0				

Вычисление функции переходов

СИМВОЛ/ СОСТОЯНИЕ	0	1	2	3	4
<i>π</i>	0	0	1	2	
A	1	1	3	1	3
B	0	2	0	4	0

Построение регулярного выражения

$(aba)^+(ba)^*(bc)?b$

Построение регулярного выражения

$(aba)^+(ba)^*(bc)?b$

1. Построить автоматы без дополнительных переходов
отдельно*

aba

ba

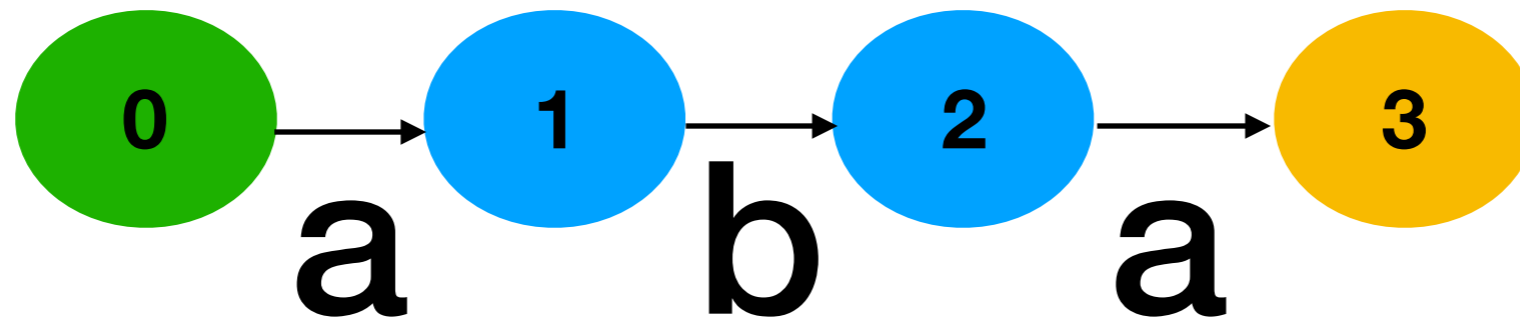
bc

b

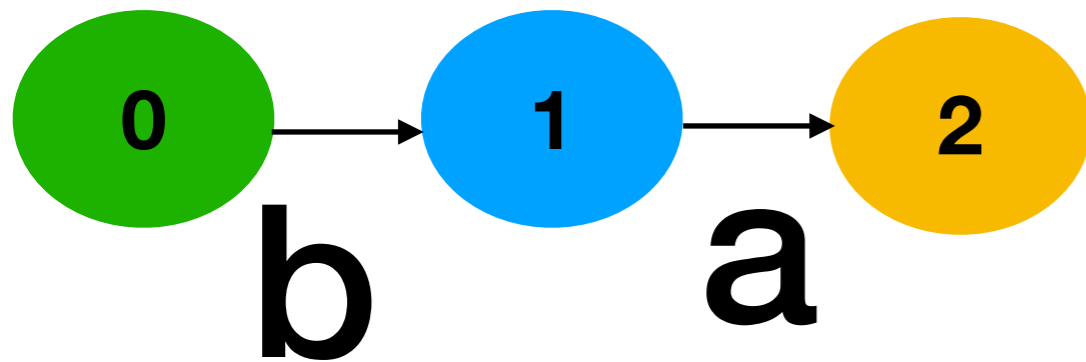
*считаем, что все переходы, кроме основных ведут к смерти состояния (без отката в 0).

Построение регулярного выражения

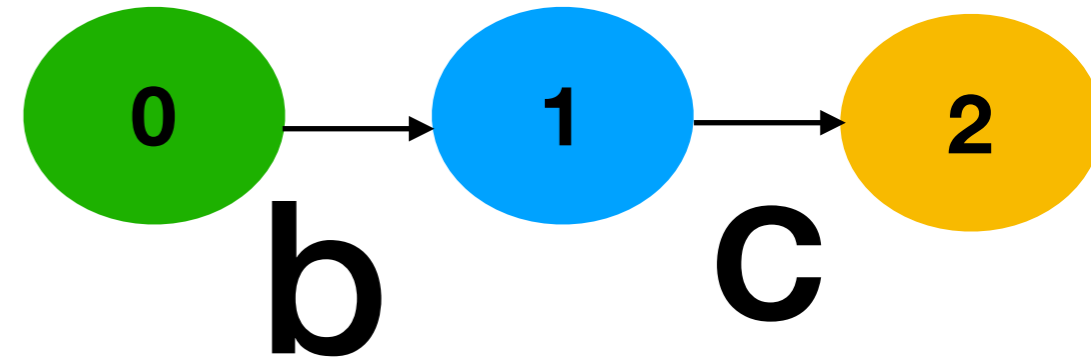
aba



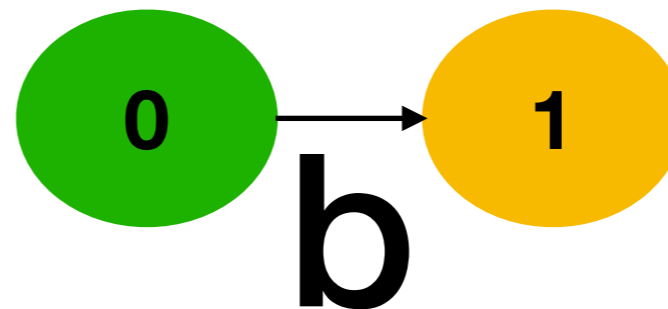
ba



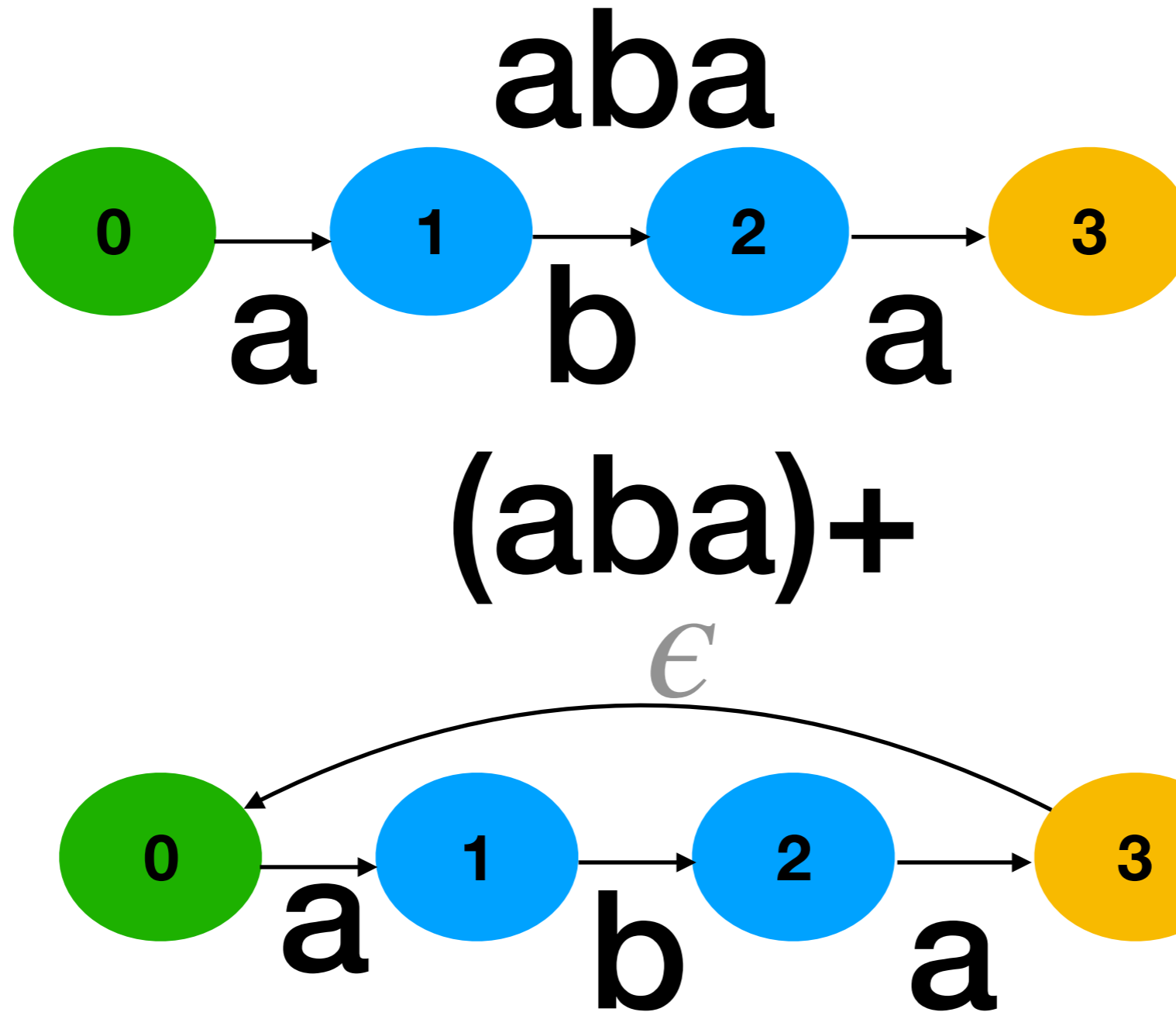
bc



b



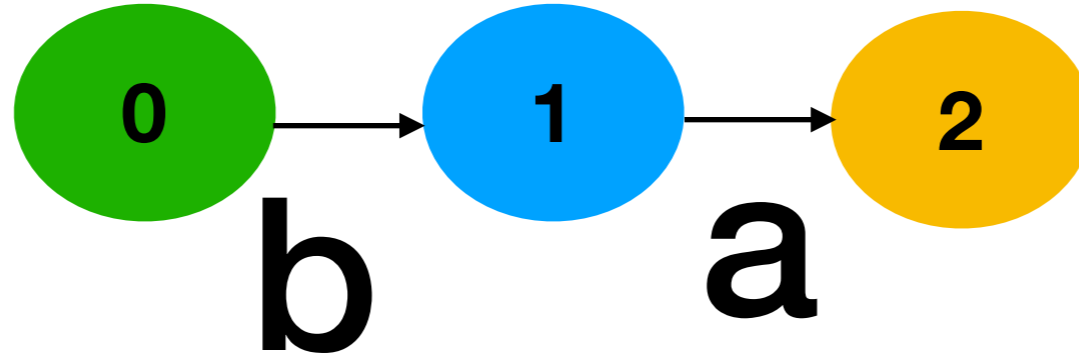
2. Добавить модификаторы



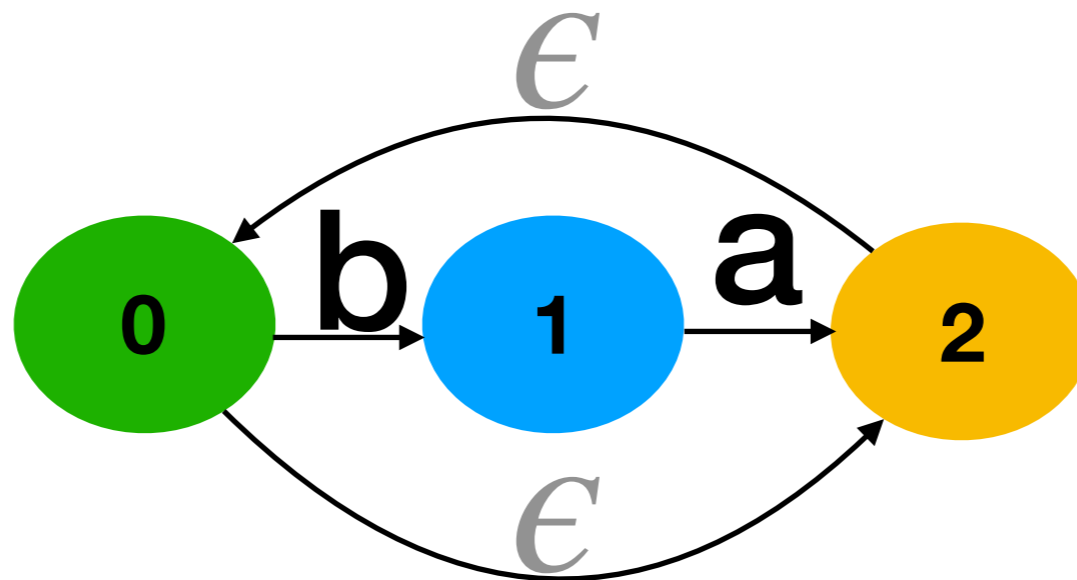
Для плюса переход по эпсилон-символу только из конечного состояния в стартовое

2. Добавить модификаторы

ba



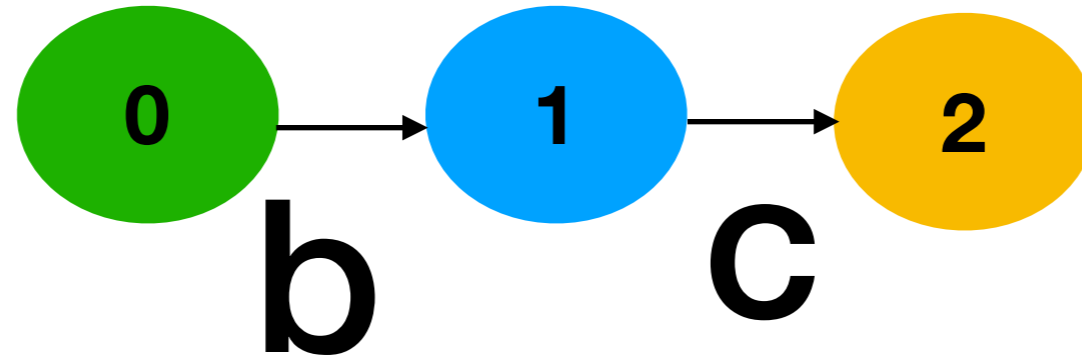
$(ba)^*$



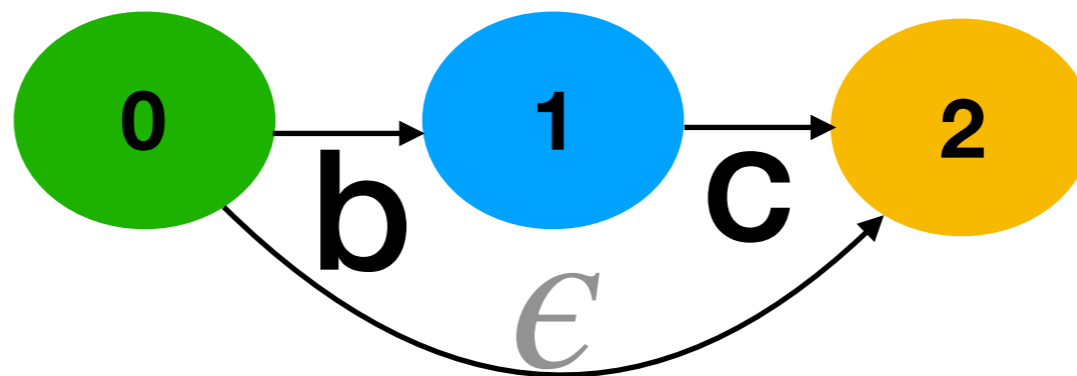
Для звездочки переход по эпсилон-символу из стартового состояния в конечное и наоборот

2. Добавить модификаторы

bc

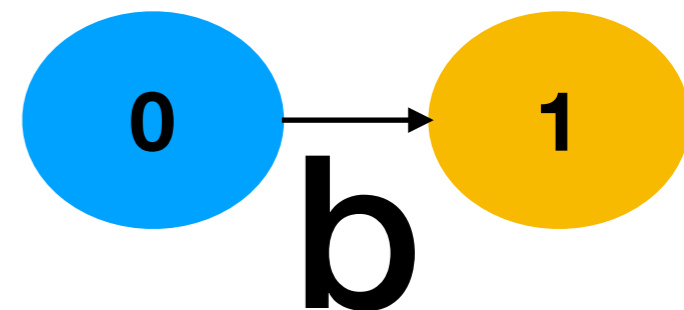
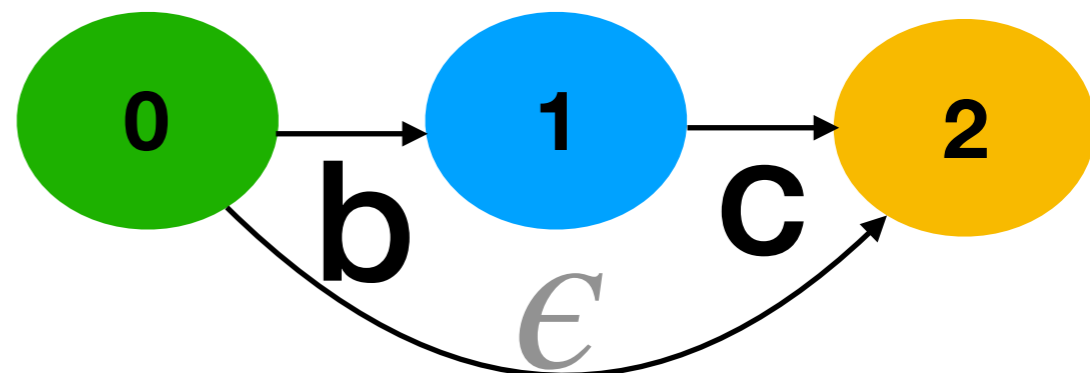
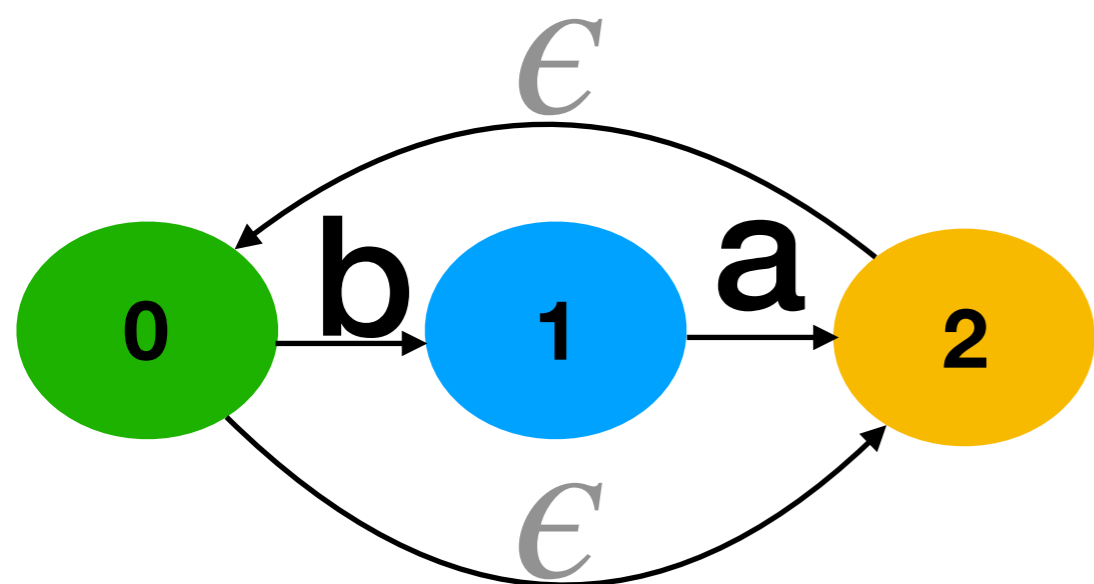
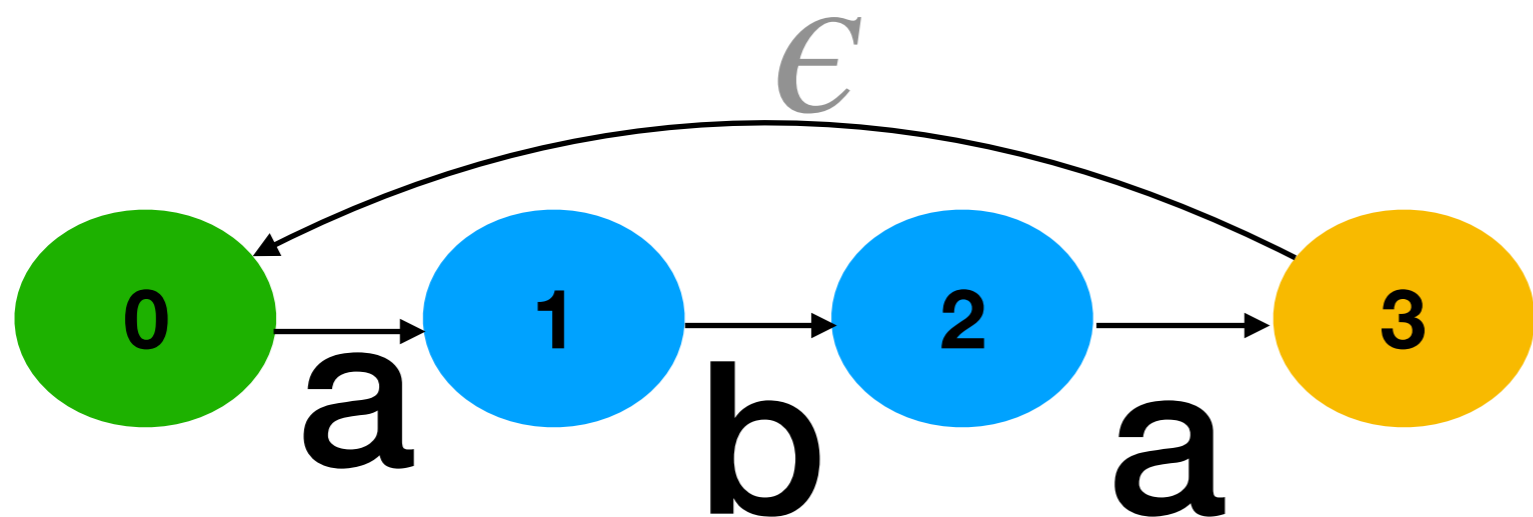


(bc)?

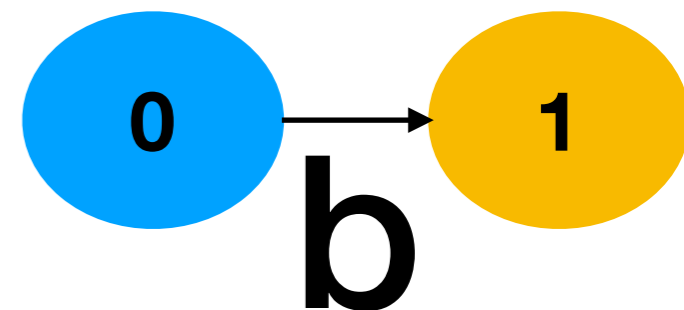
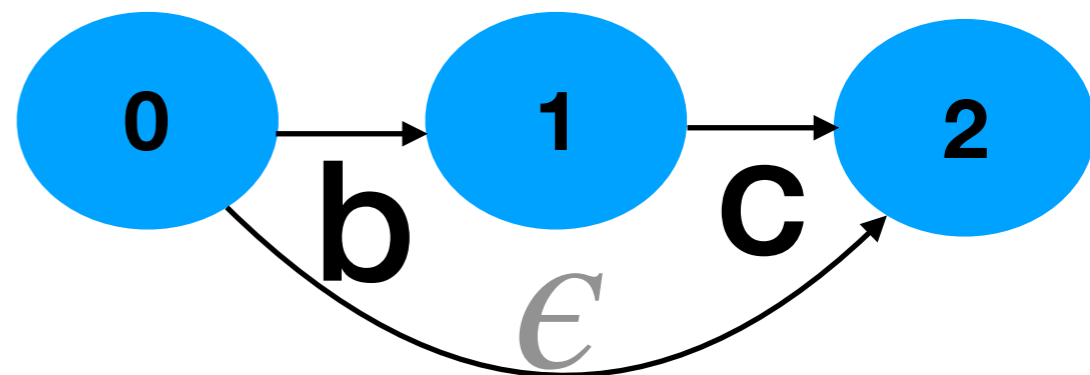
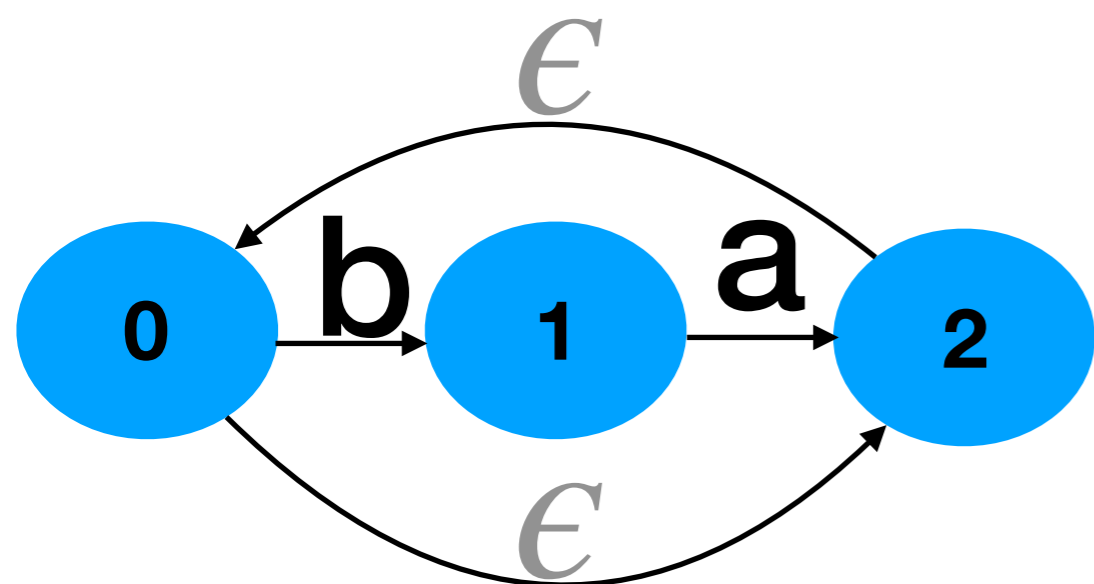
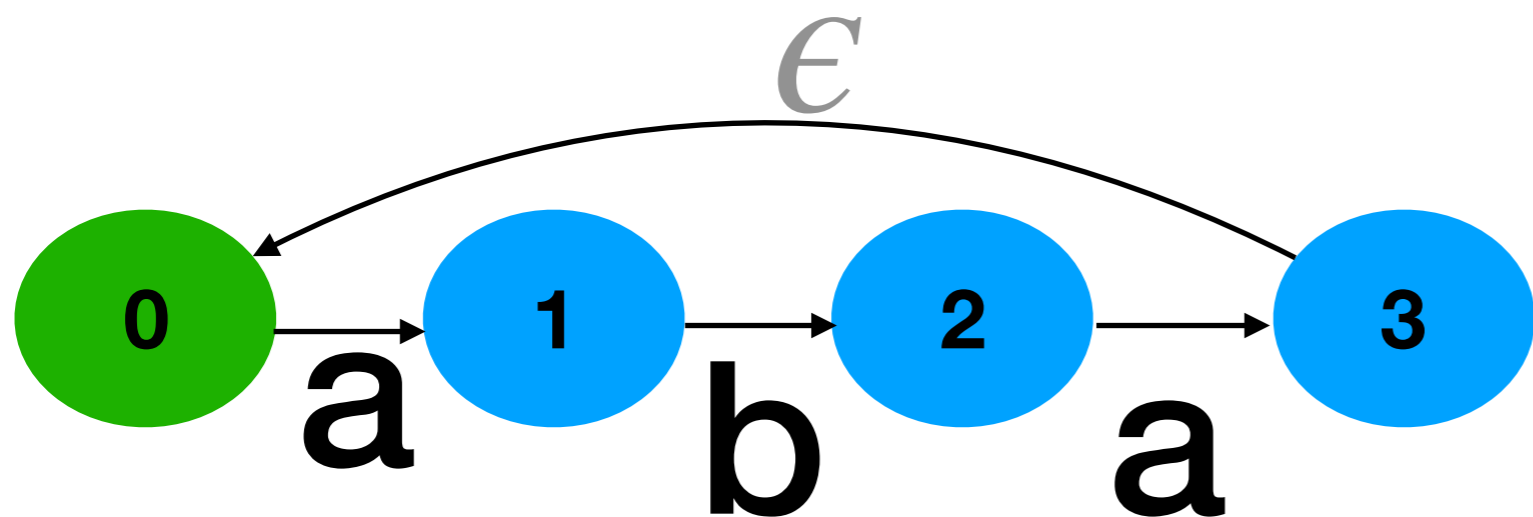


Для вопроса (0 или 1 вхождение) - добавить переход по эпсилон-символу из стартового состояния в конечное и наоборот

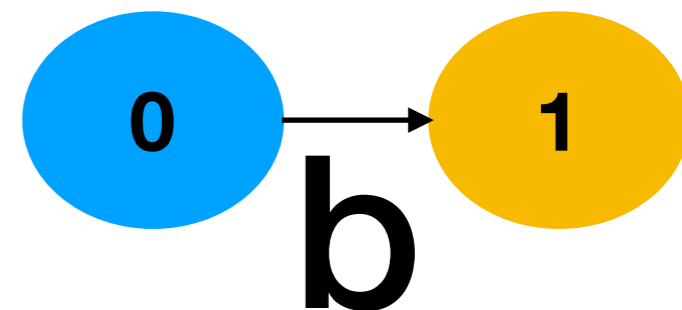
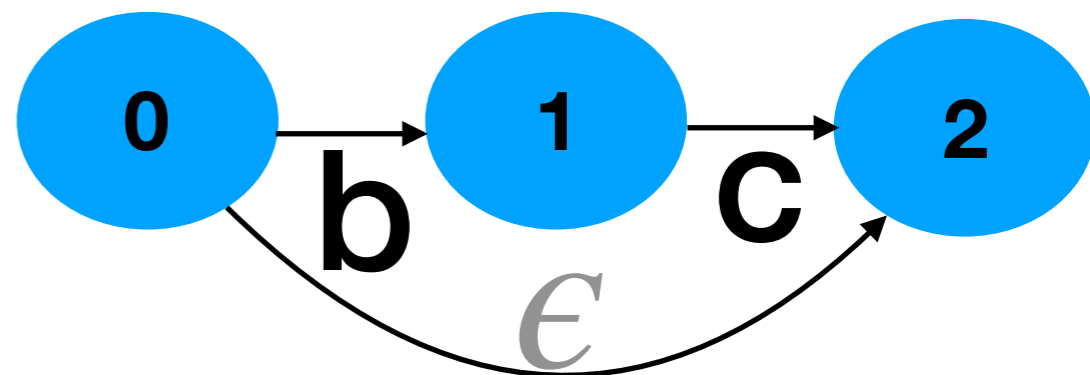
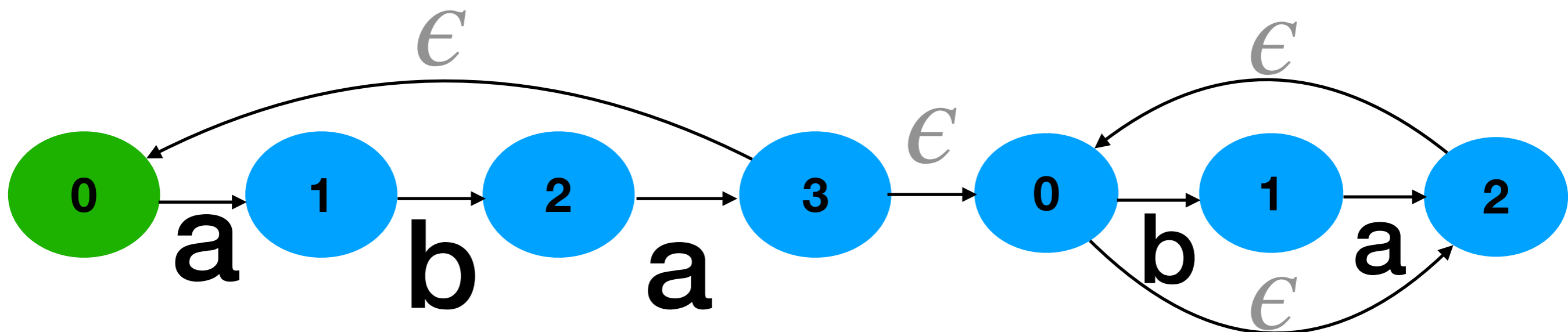
3. Соединить автоматы, стартовое состояние остается стартовым только у первого, конечное конечным - только у последнего



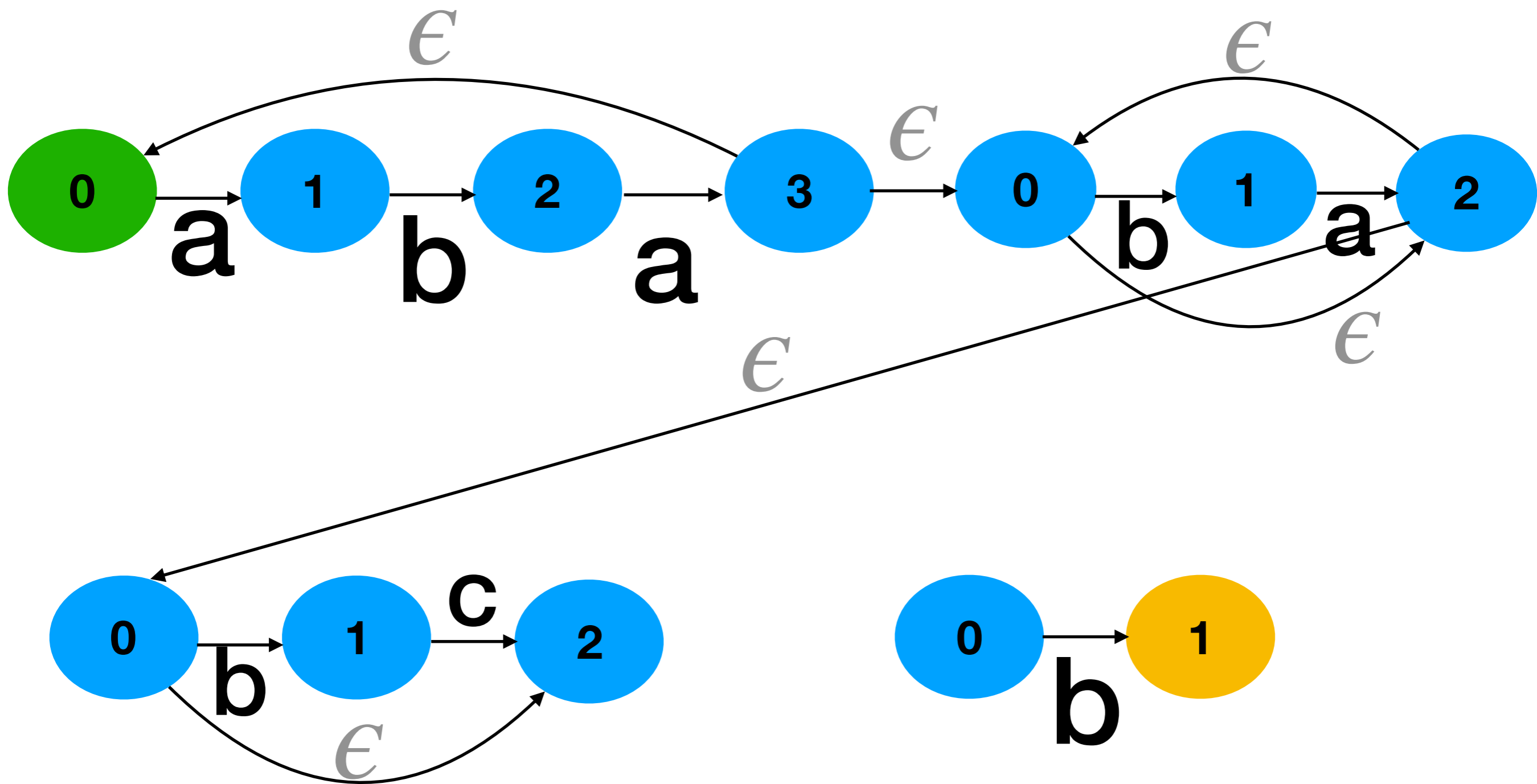
3. Соединить автоматы, стартовое состояние остается стартовым только у первого, конечное конечным - только у последнего



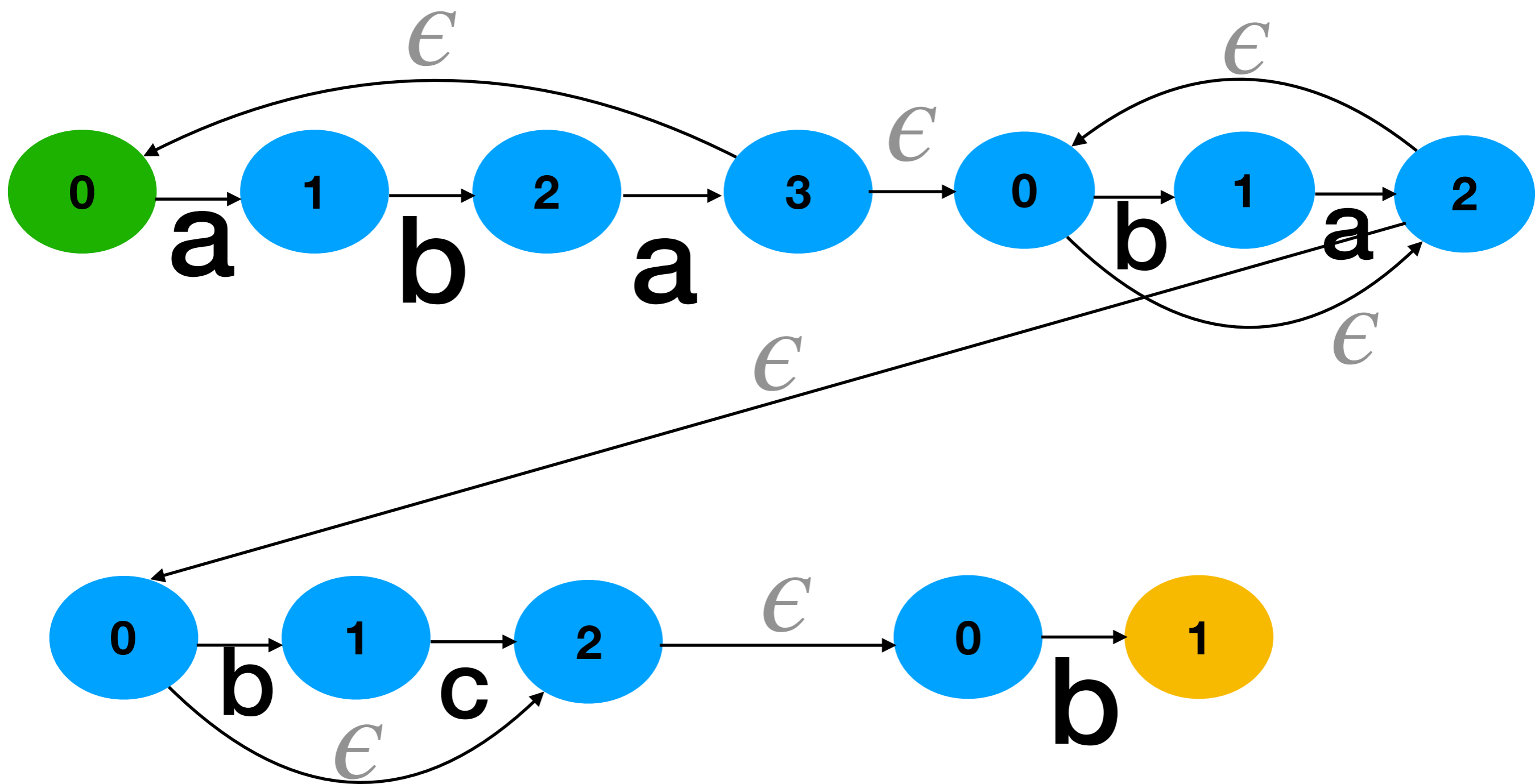
3. Соединить автоматы, стартовое состояние остается стартовым только у первого, конечное конечным - только у последнего



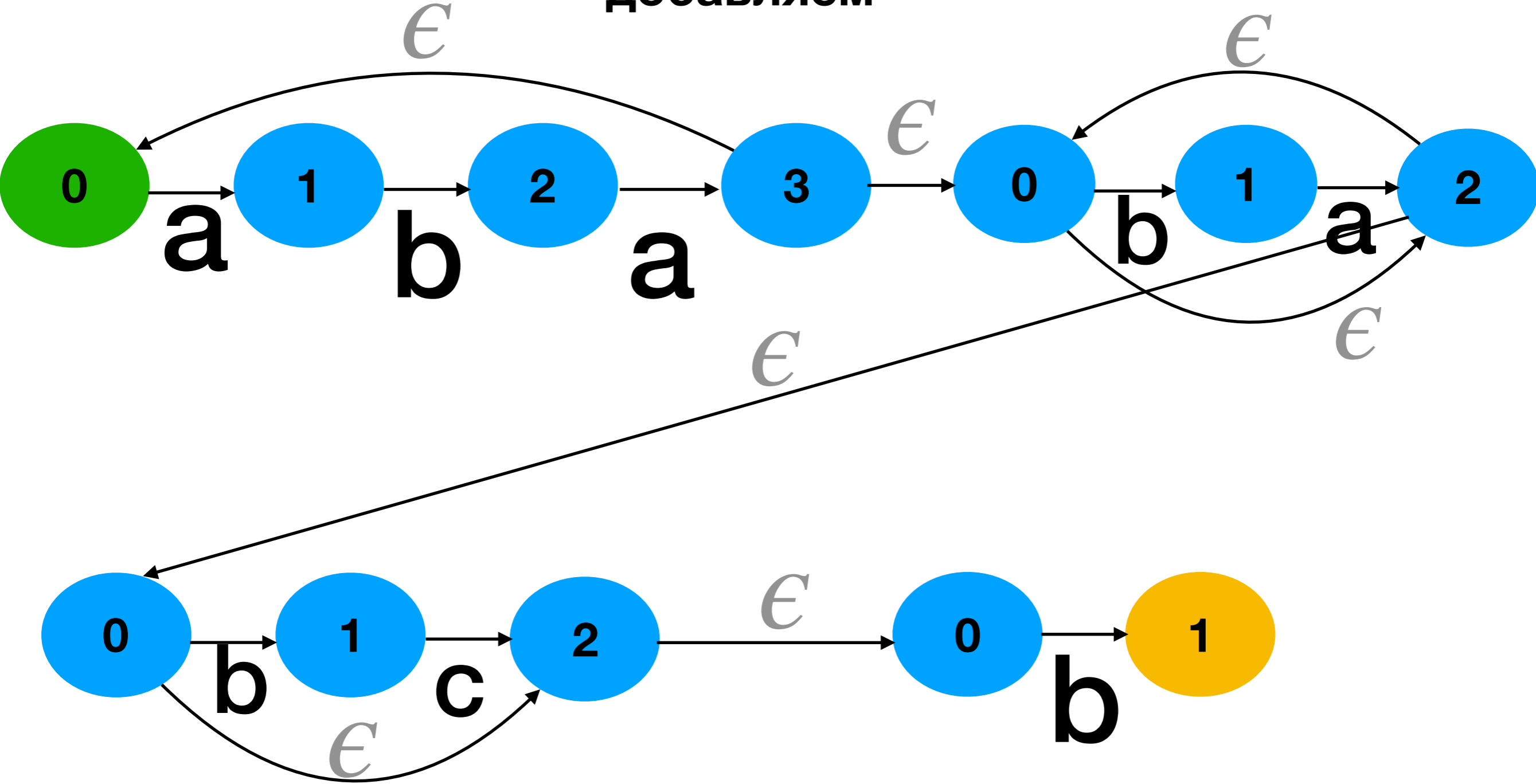
3. Соединить автоматы, стартовое состояние остается стартовым только у первого, конечное конечным - только у последнего



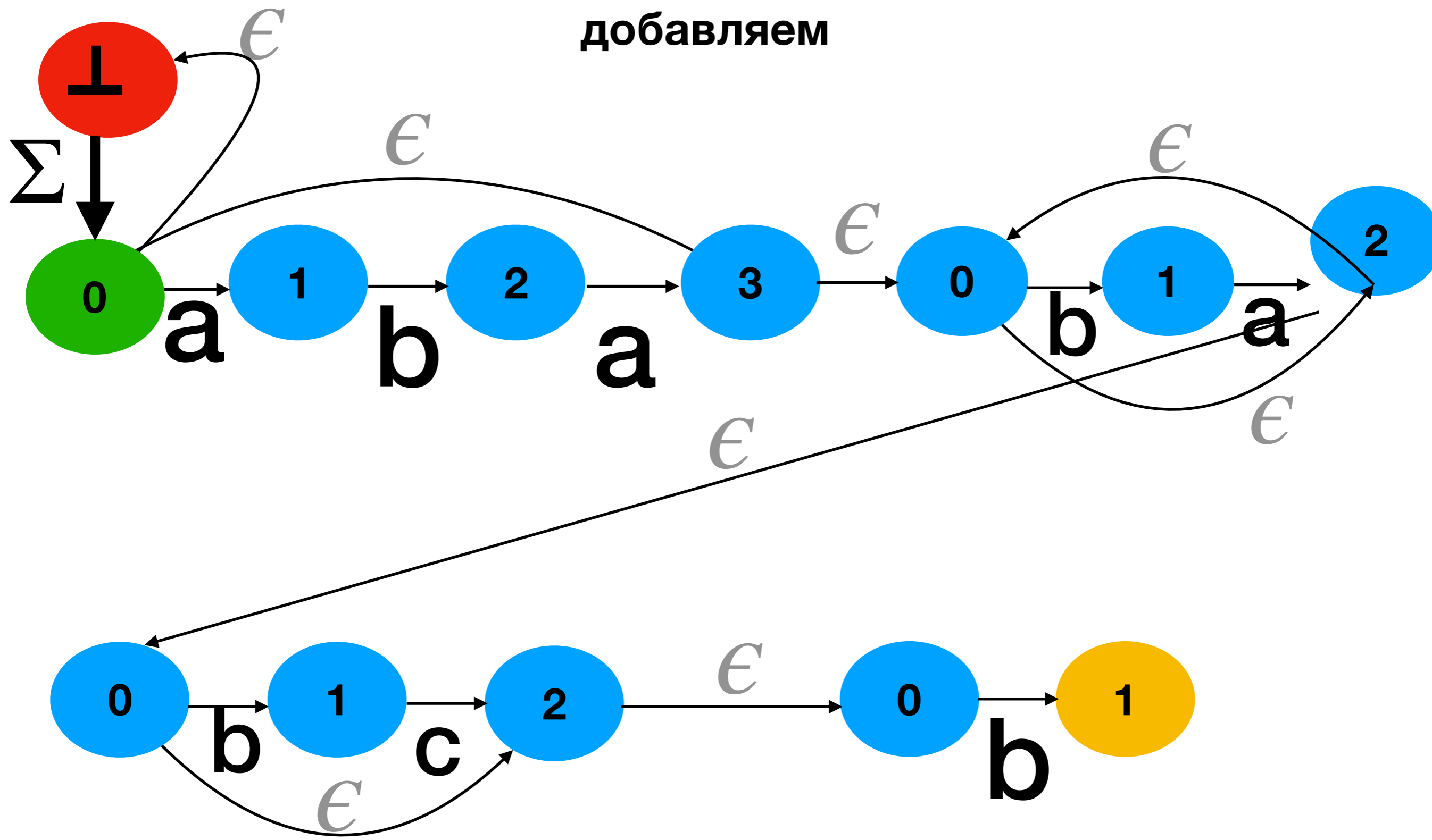
3. Соединить автоматы, стартовое состояние остается стартовым только у первого, конечное конечным - только у последнего



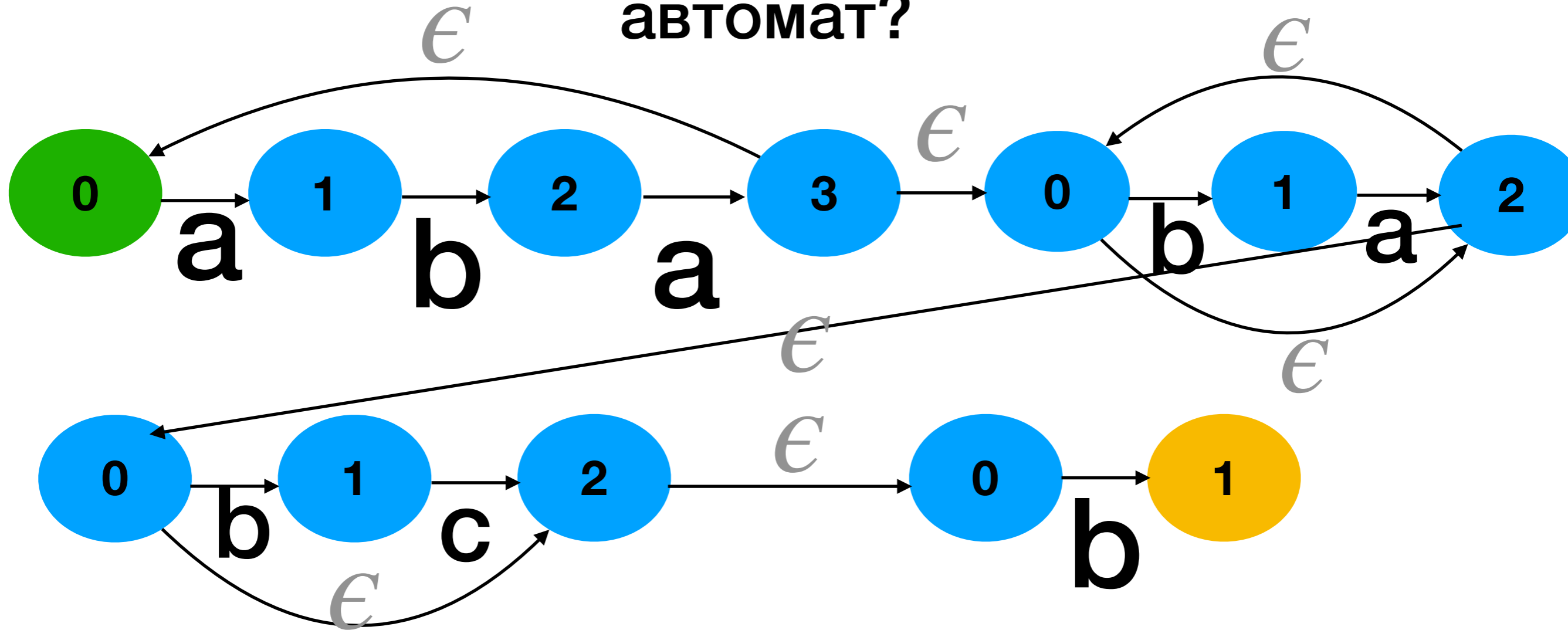
4. Если хотим искать паттерн во всем тексте - добавить в стартовом состоянии эпсилон переход в само себя. Если проверяем, что в начале текста есть паттерн - то ничего не добавляем



4. Если хотим искать паттерн во всем тексте - добавить состояние \perp с переходом в стартовое состояние по любому символу и из 0 добавить эпсилон-переход в \perp . Если проверяем, что в начале текста есть паттерн - то ничего не добавляем

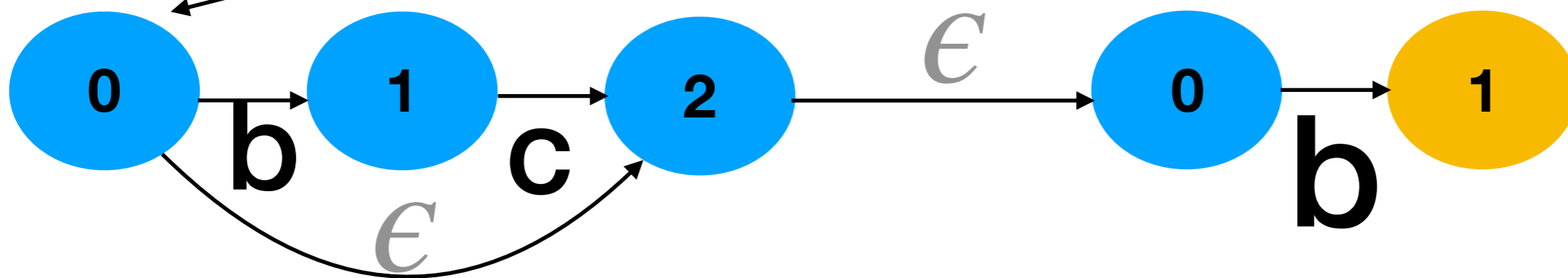
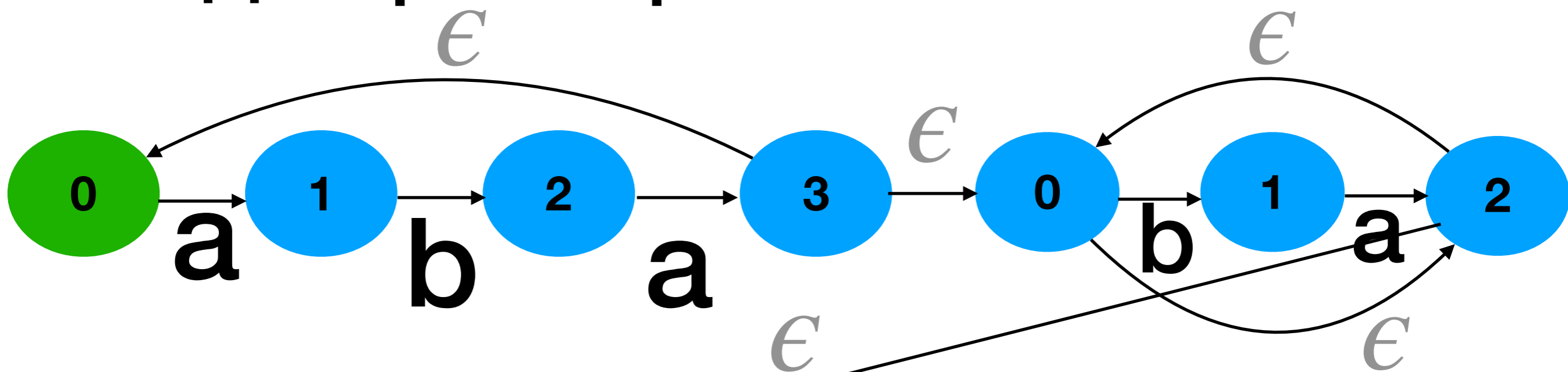


Сколько построить недетерминированный автомат?



Всего нужно добавить максимально по 2 состояния на один символ + (возможно, 1 вечно живое состояние), на каждый символ в паттерн добавляется максимум два перехода (1 переход добавляется для символов алфавита и + и ?, 2 перехода - для звезды) + на каждую часть паттерна еще не более M соединяющих эпсилон переходов

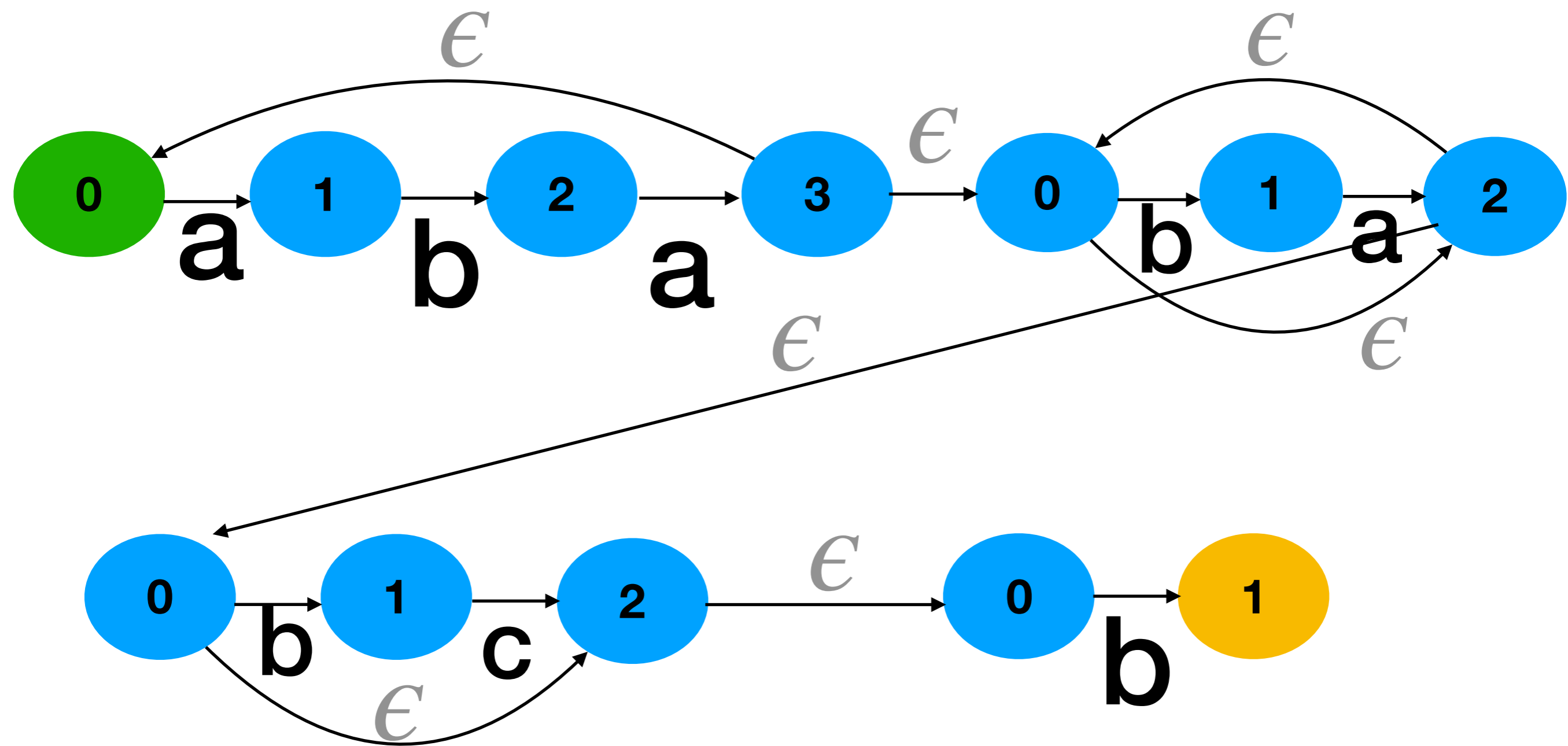
Сколько строит недетерминированный автомат?



$$O(1) + O(2 * M) + O(2 * M) + O(M) = O(M)$$

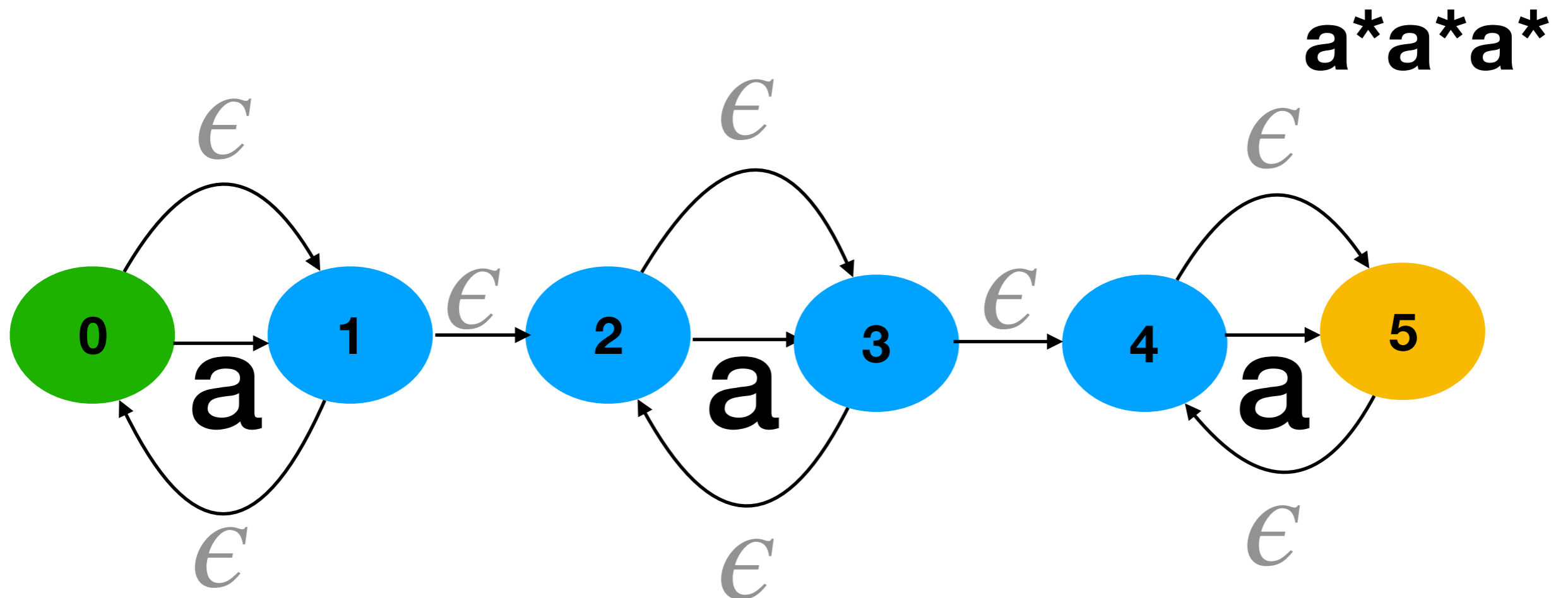
времени

Сколько проверить недетерминированный автомат, что последовательность удовлетворяет паттерну?



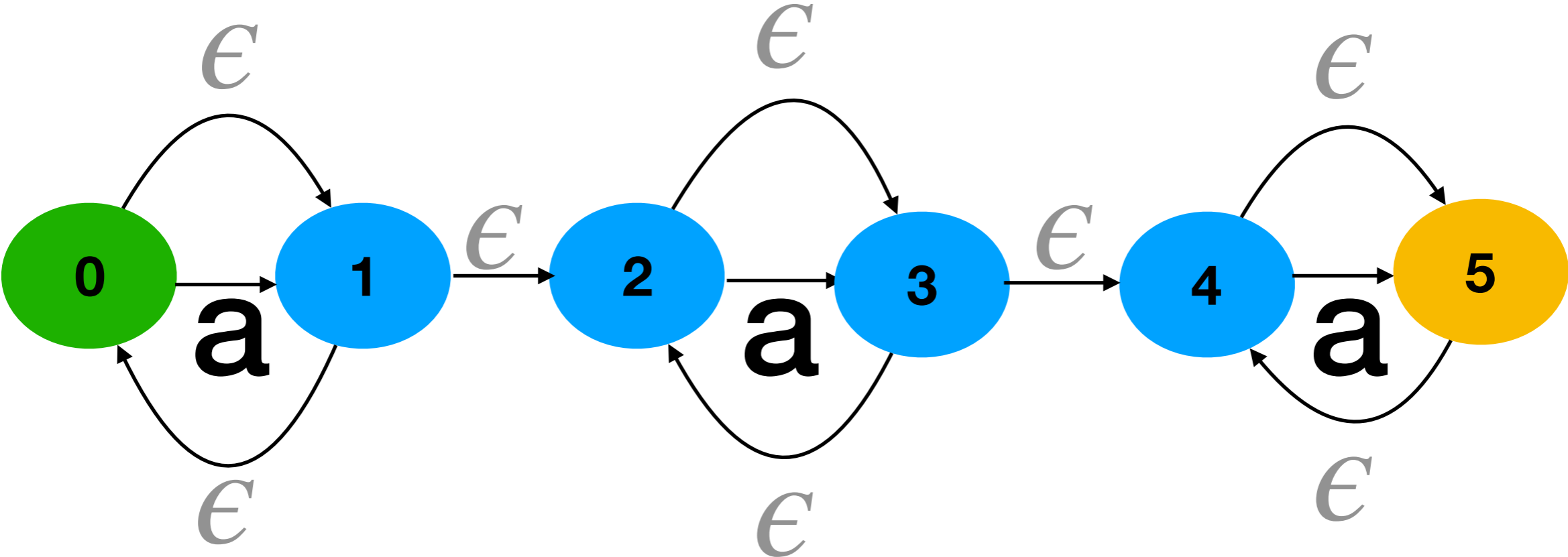
Сколько проверять недетерминированный автомат, что в последовательности есть паттерн?

В худшем случае у нас есть по два эпсилон-перехода на каждый спец-символ + порядка соединяющих M переходов + переходы по регулярным символам (не более M). Значит, всего - не более $O(M)$ переходов

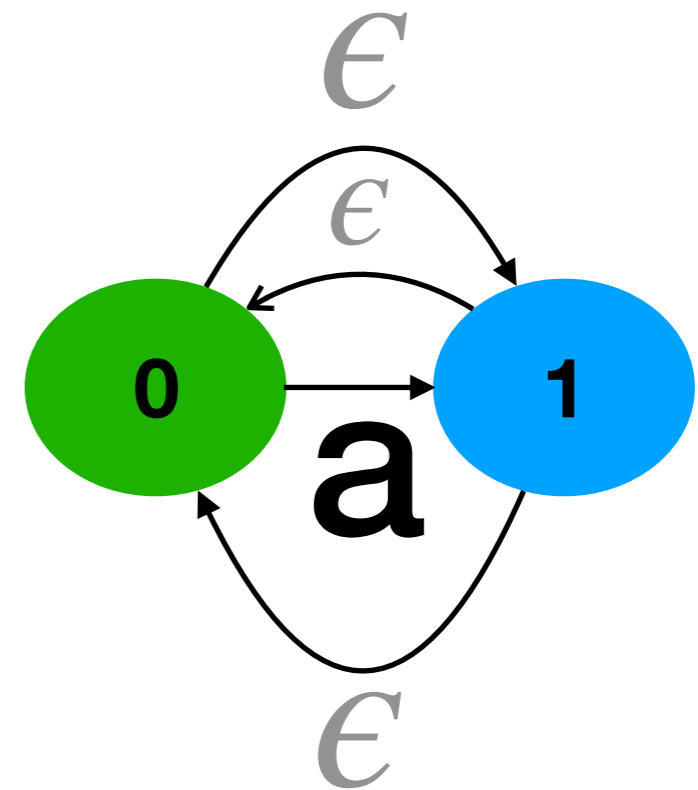
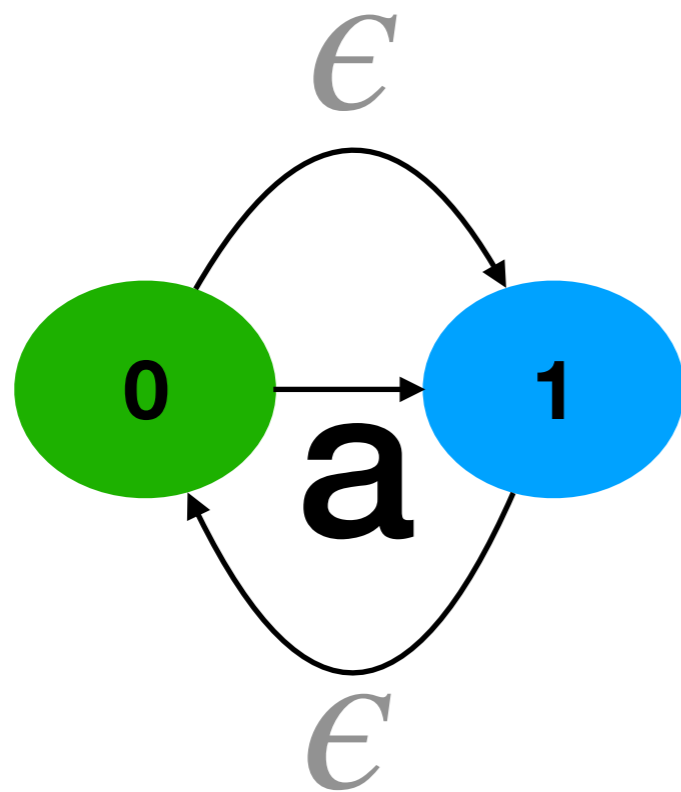


В данном случае состояния с одинаковыми номерами для нас неразличимы

Например, состояние 1 может быть получено как эпсилон переходом из состояния 0, так и прочтением символа а в состоянии 0

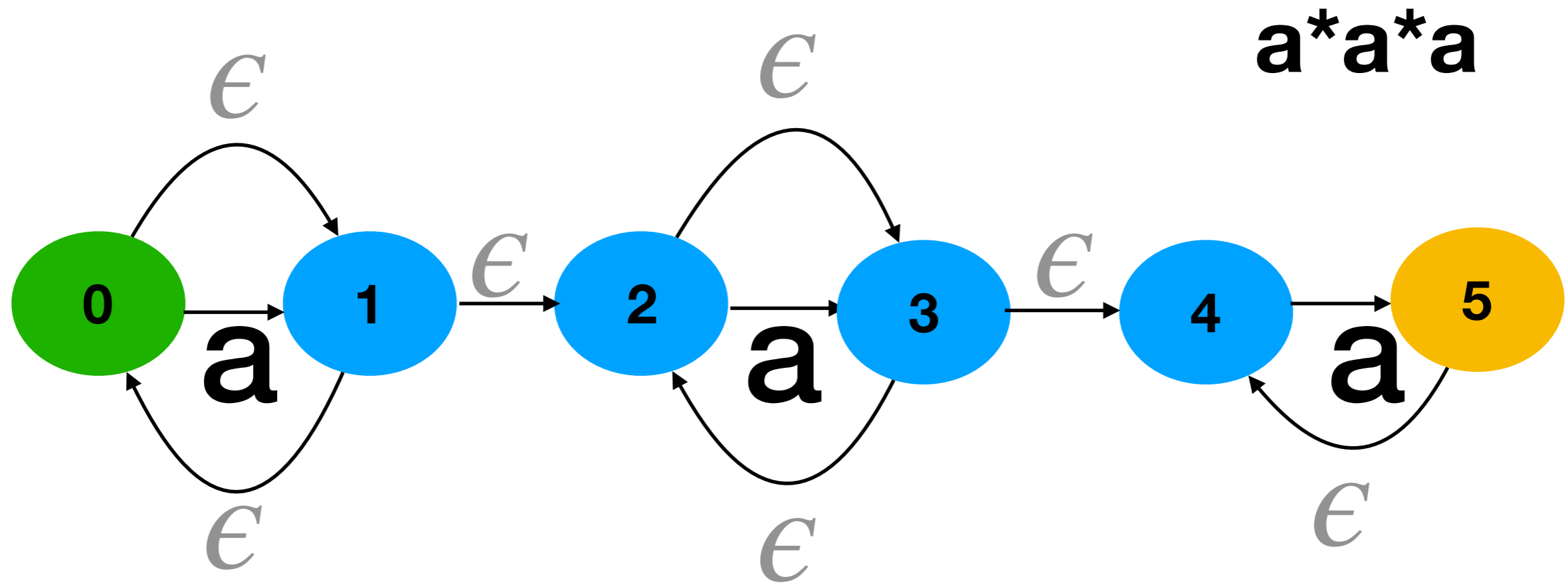


Так как состояния неразличимы то автомат $(a^*)^+$ не отличается для нас от автомата a^*



Поэтому мы можем считать только разные стрелки

В любой момент времени у нас могут быть активны в худшем случае все состояния



Например, здесь за счет эpsilon переходов после прочтения символа “a” будут активны все состояния, кроме конечного

Сколько проверить недетерминированный автомат, что в последовательности есть паттерн?

В худшем случае у нас не более $O(M)$ переходов всего в
автомате

В любой момент времени у нас могут быть активны в худшем случае
все состояния

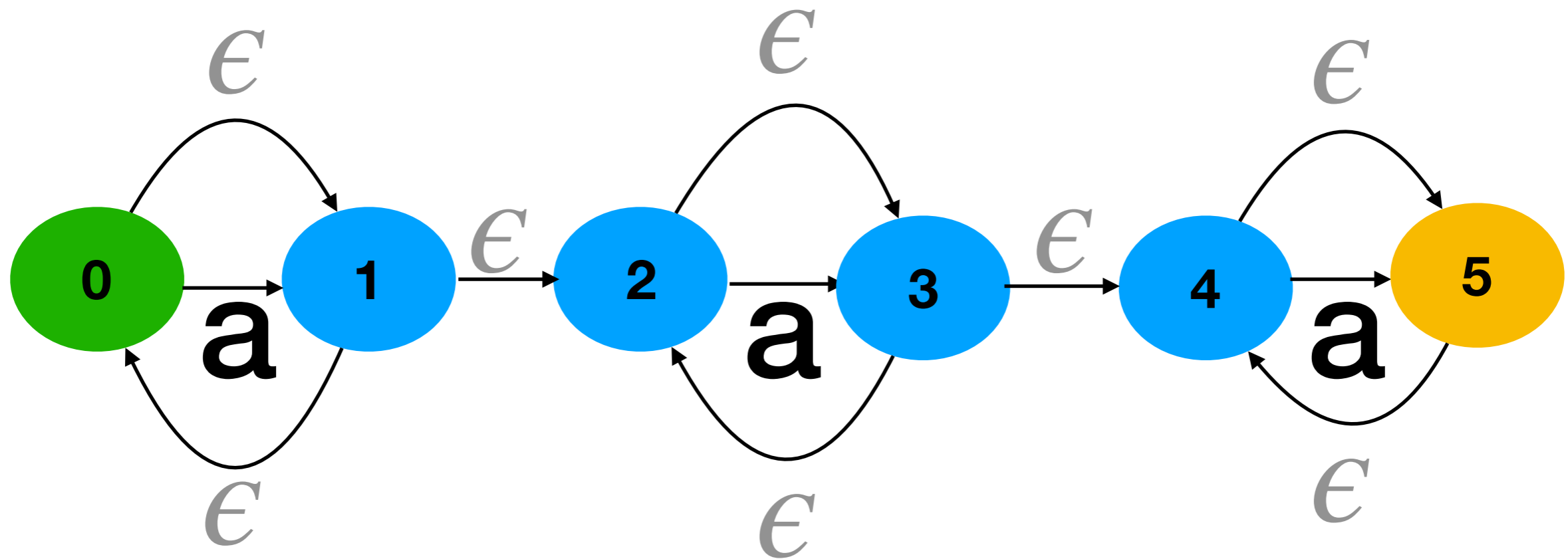
Таким образом, при прочтении одного символа автомат осуществит
максимально $O(M)$

Всего символов в тексте N . Тогда суммарная сложность автомата:

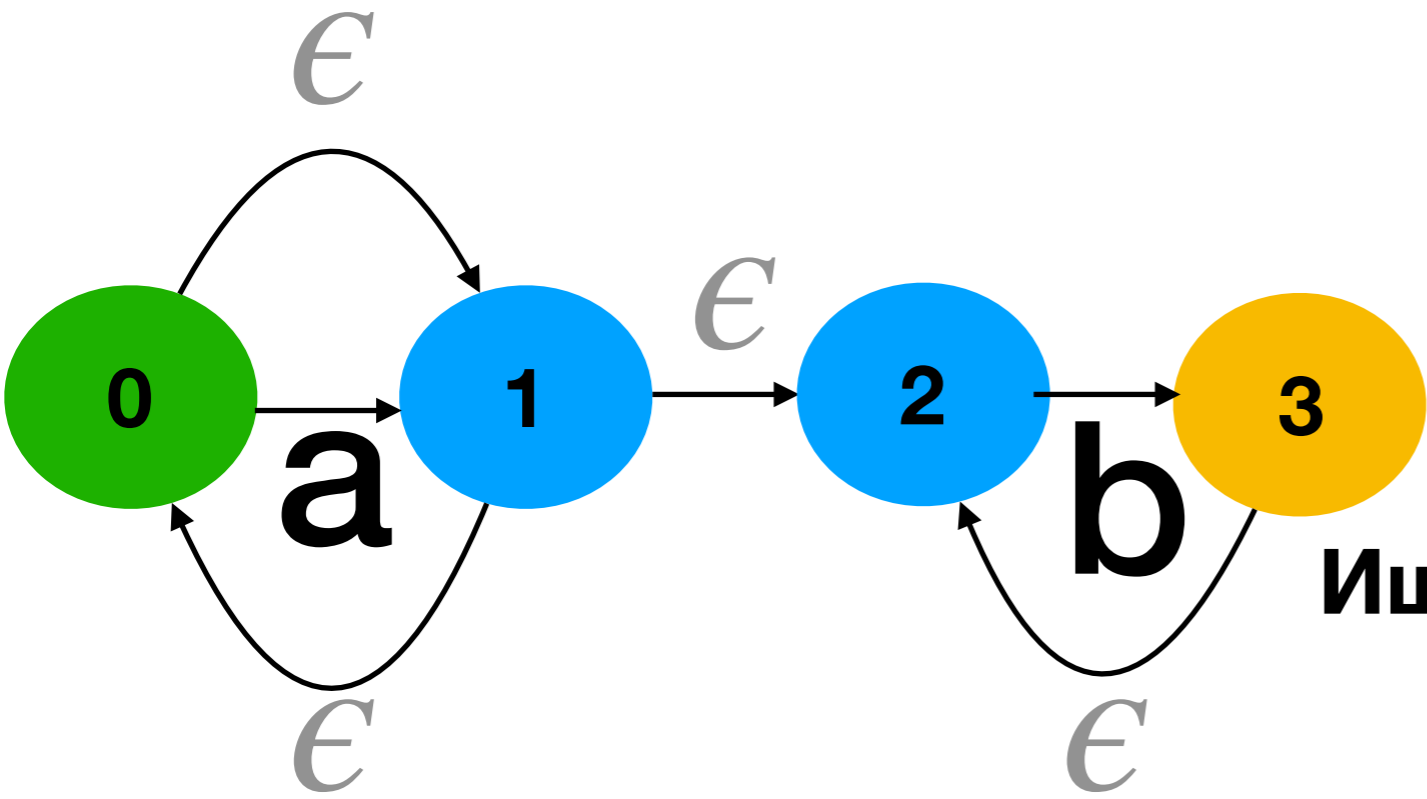
$$O(M) \cdot N = O(MN)$$

Сколько исكاتь недетерминированным автоматом в последовательности наиболее длинное/короткое вхождение паттерн?

Проведем аналогичные рассуждения. На каждом этапе мы попадаем в одно состояние несколькими путями



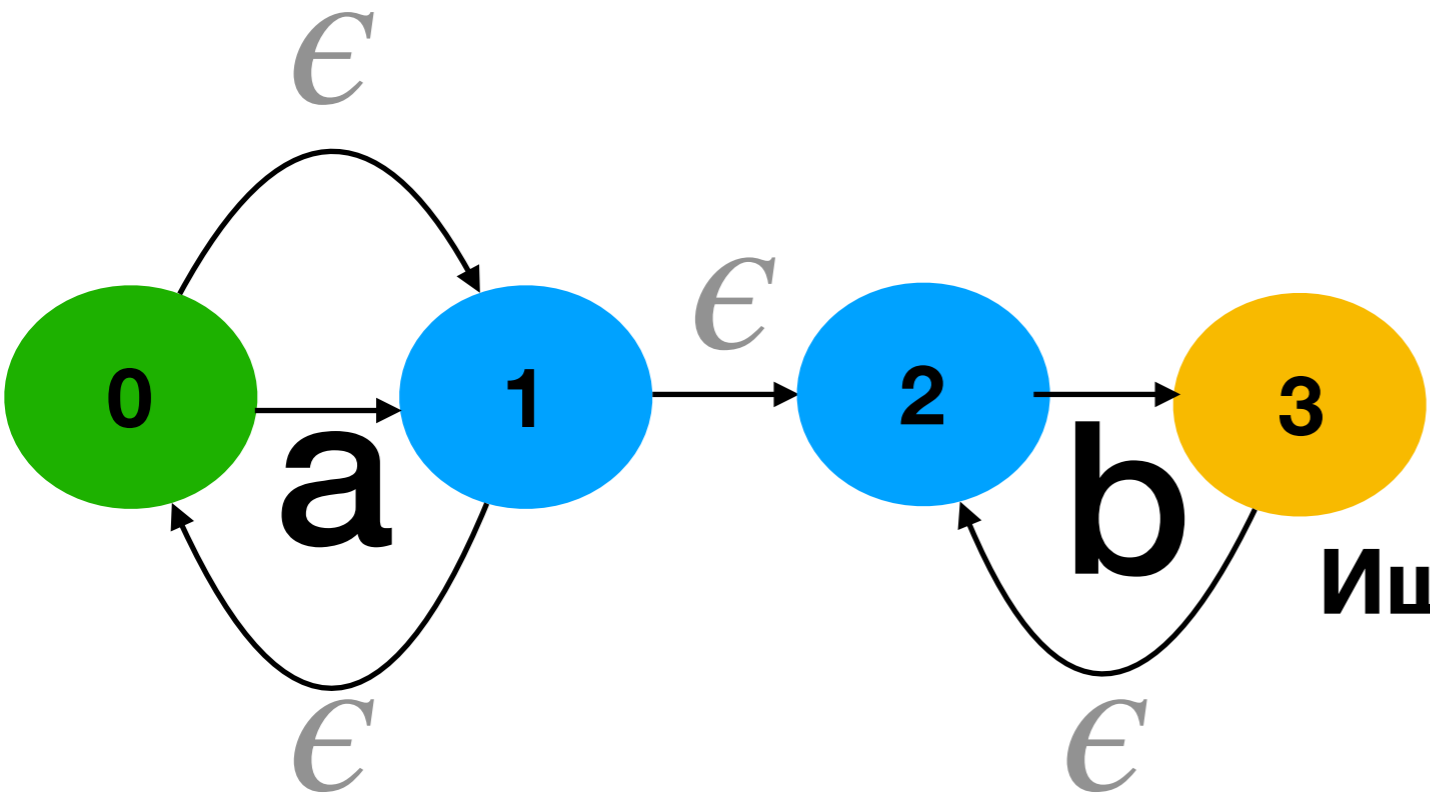
a*b+



Ищем паттерн в строке aabb

Символ текста	ϵ	a	ϵ	a	ϵ	b	ϵ	b	ϵ
Начинае									
М В									
0	0(“”) 1(“”) 2(“”)	1(“a”)							

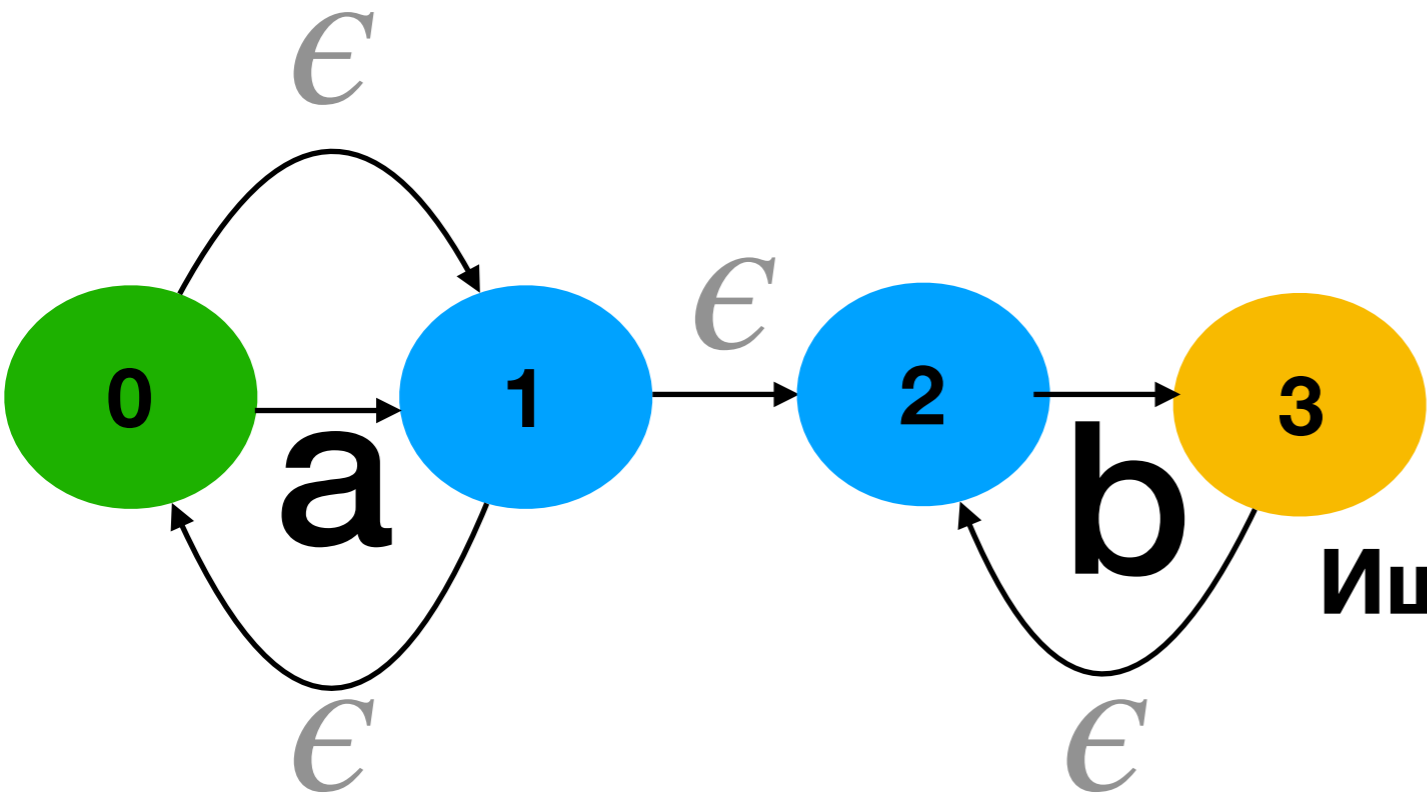
a*b+



Ищем паттерн в строке aabb

Символ текста	ϵ	a	ϵ	a	ϵ	b	ϵ	b	ϵ
Начинае M B 0	0(" ") 1(" ") 2(" ")	1("a")	0("a") 1("a") 2("a")						

a*b+

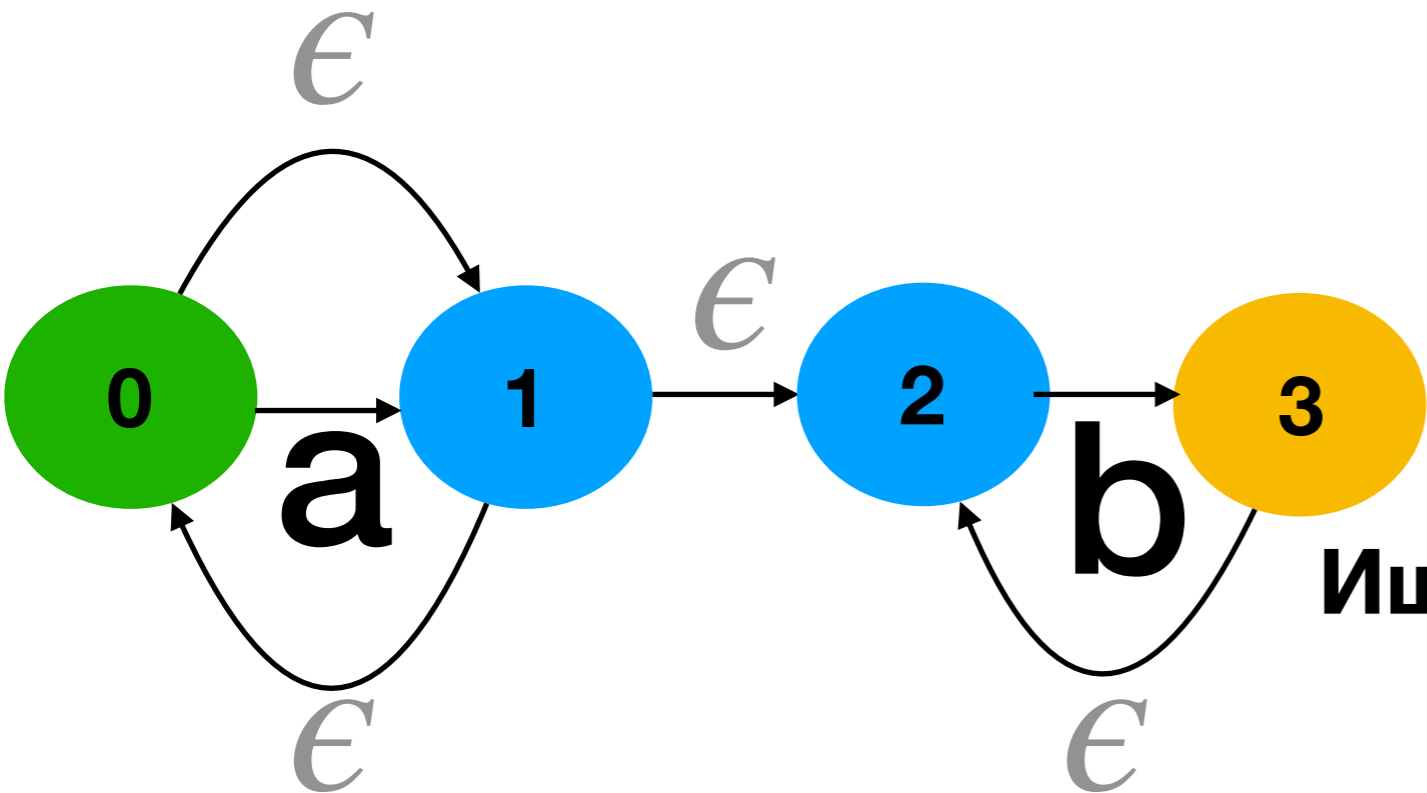


Ищем паттерн в строке aabb

3 - состояние конечное, даже если нет перехода из него - остается жить

Символ текста	ε	a	ε	a	ε	b	ε	b	ε
Начинаем в 0	0(“”) 1(“”) 2(“”)	1(“a”)	0(“a”) 1(“a”) 2(“a”)	1(“aa”)	0(“aa”) 1(“aa”) 2(“aa”)	3(“aab”)	3(“aab”) 2(“aab”)	3(“aabb”) 3(“aab”)	Новых конечных состояний не появляется

a*b+



Ищем паттерн в строке aabb

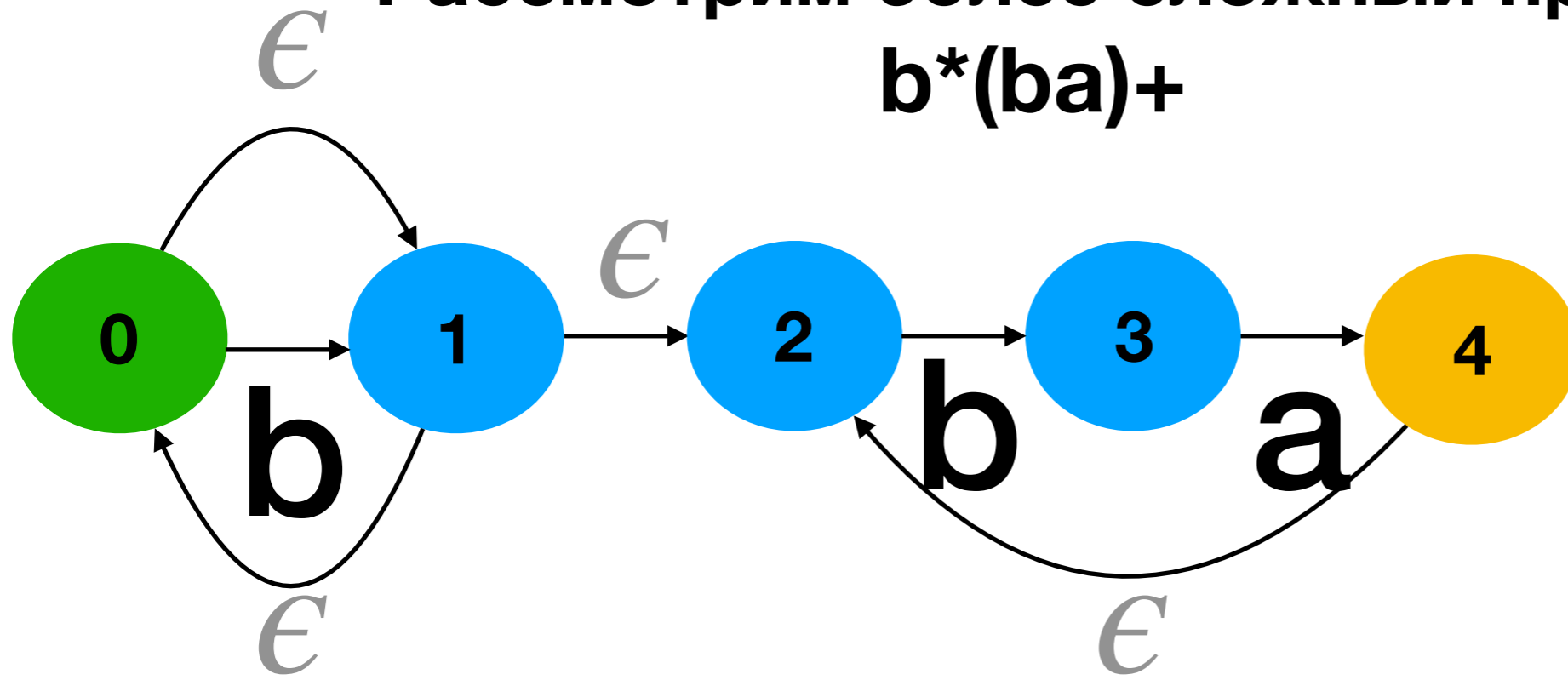
Выбираем состояние с наибольшей длиной вхождения

Символ текста	ε	a	ε	a	ε	b	ε	b	ε
Начинае M B 0	0(“”) 1(“”) 2(“”)	1(“a”)	0(“a”) 1(“a”) 2(“a”)	1(“aa”)	0(“aa”) 1(“aa”) 2(“aa”)	3(“aab”)	3(“aab”) 2(“aab”)	3(“aabb”) 3(“aab”)	3(“aab b”)

Заметим, что и в этом случае автомат $(a^*)_+$ можно упростить до a^*

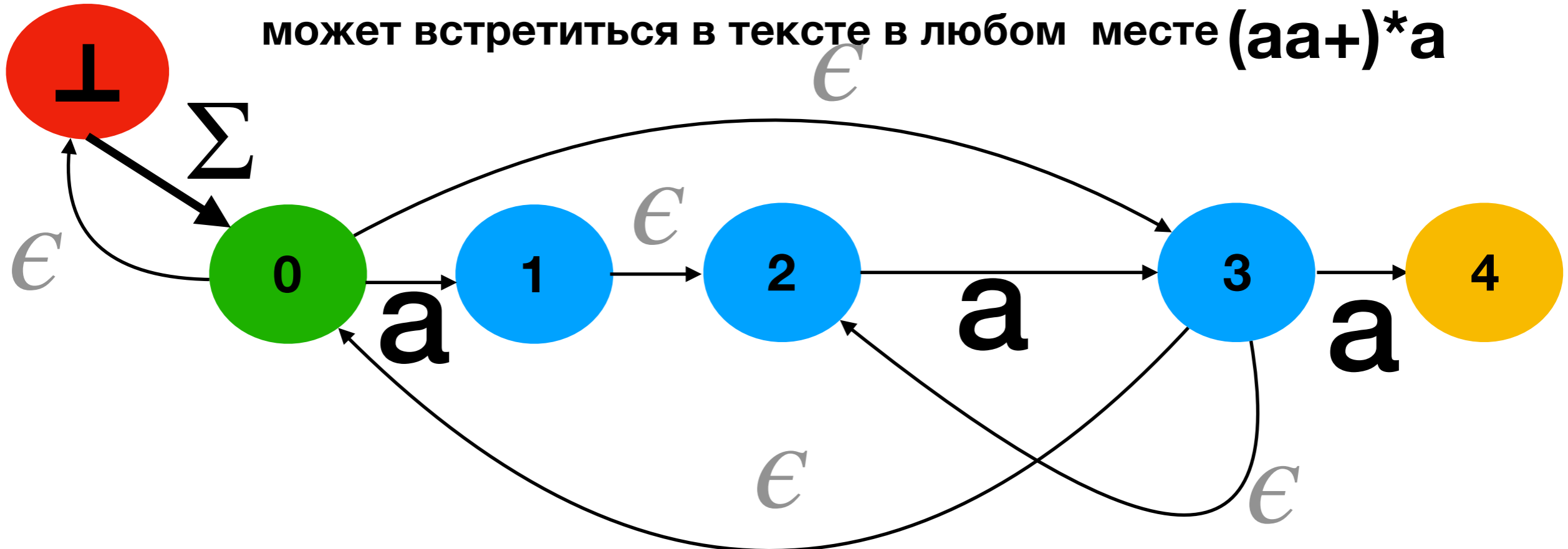
Рассмотрим более сложный пример

$b^*(ba)^+$



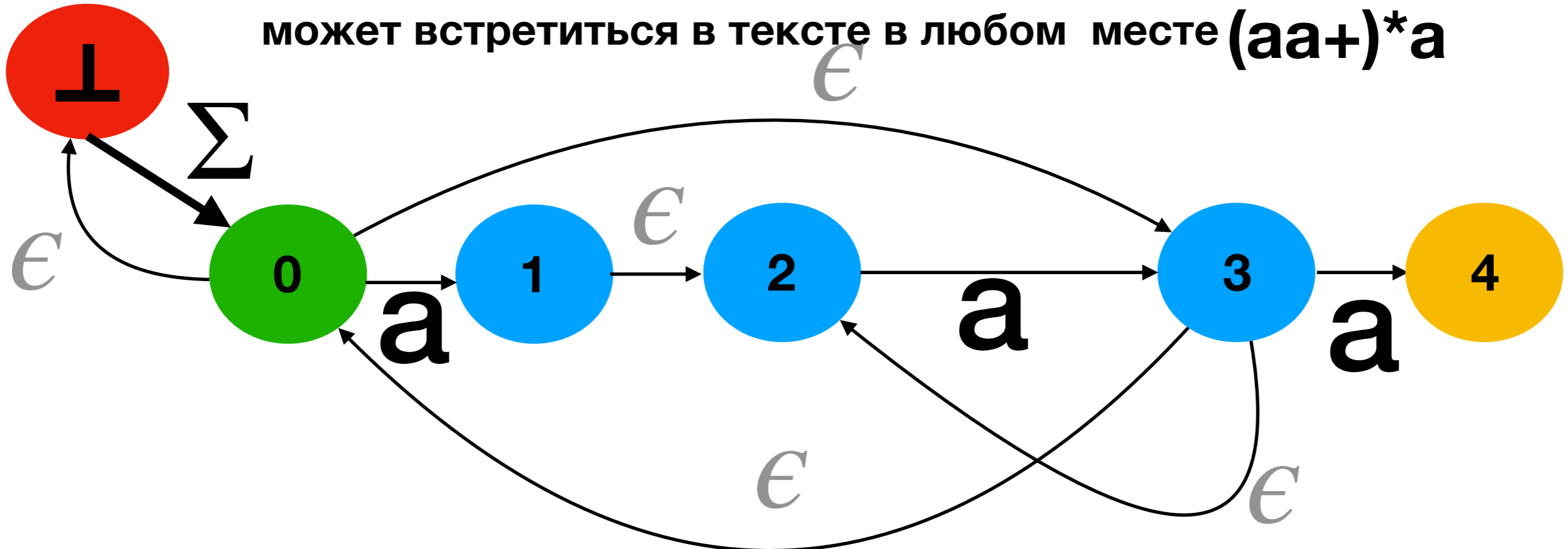
Символ текста	ϵ	b	ϵ	b	ϵ	b	ϵ	a	ϵ
Начинаем									
В	0(“”)	1(“b”)	0(“b”)	1(“bb”)	0(“bb”)	1(“bbb”)	0(“bbb”)		4(“bbb
0	1(“”)	3(“b”)	1(“b”)	3(“bb”)	1(“bb”)	3(“bbb”)	1(“bbb”)	4(“bbba”)	a”)
	2(“”)		2(“b”)		2(“bb”)		2(“bbb”)		2(“bbb
			3(“b”)		3(“bb”)		3(“bbb”)		a”)

Рассмотрим и еще более сложный пример, мы предполагаем, что паттерн может встретиться в тексте в любом месте $(aa+)^*a$



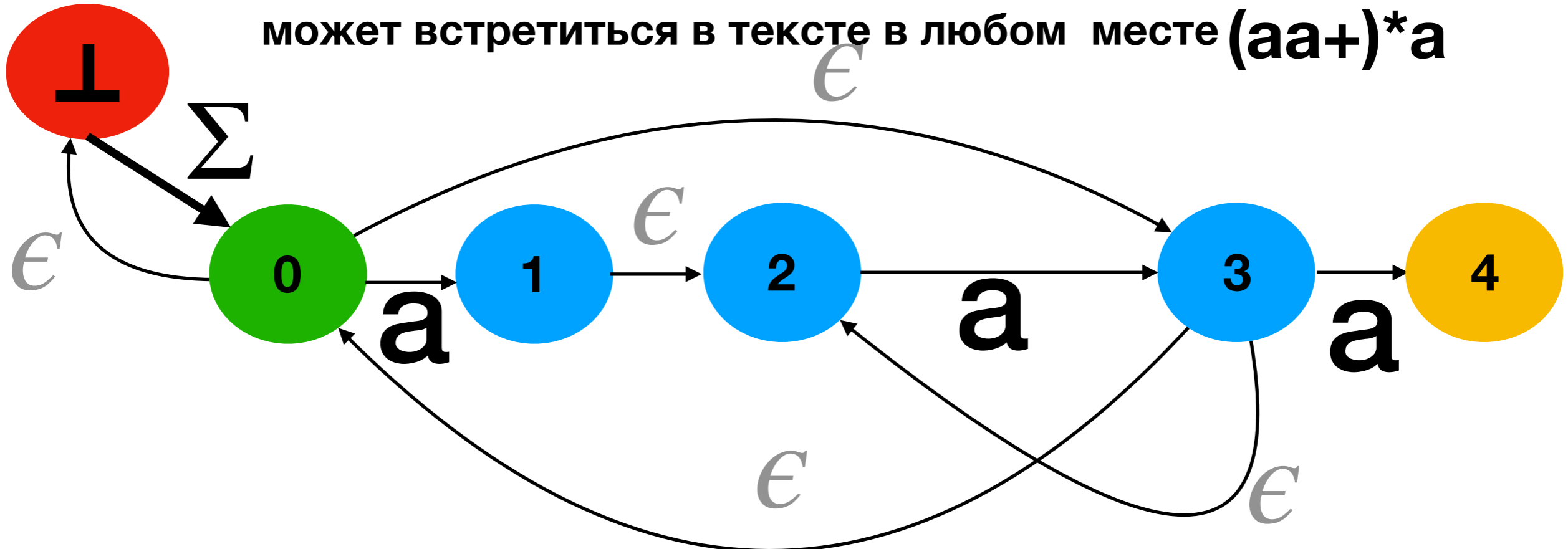
Символ текста	ϵ	a	ϵ	a	ϵ	a	ϵ
Начинаем в 0	0(“”) 3(“”) 2(“”) ⊥	1(“a”) 4(“a”) 3(“a”) 0(“-”)	1(“a”) 2(“a”) 4(“a”) 3(“a”) 0(“a”), 0(“-”) 3(“-”), 2(“-”) ⊥				“-” - обозначает, что вхождение паттерна не в начале (как гэп), в длину вхождения вклад не вносит

Рассмотрим и еще более сложный пример, мы предполагаем, что паттерн может встретиться в тексте в любом месте $(aa+)^*a$



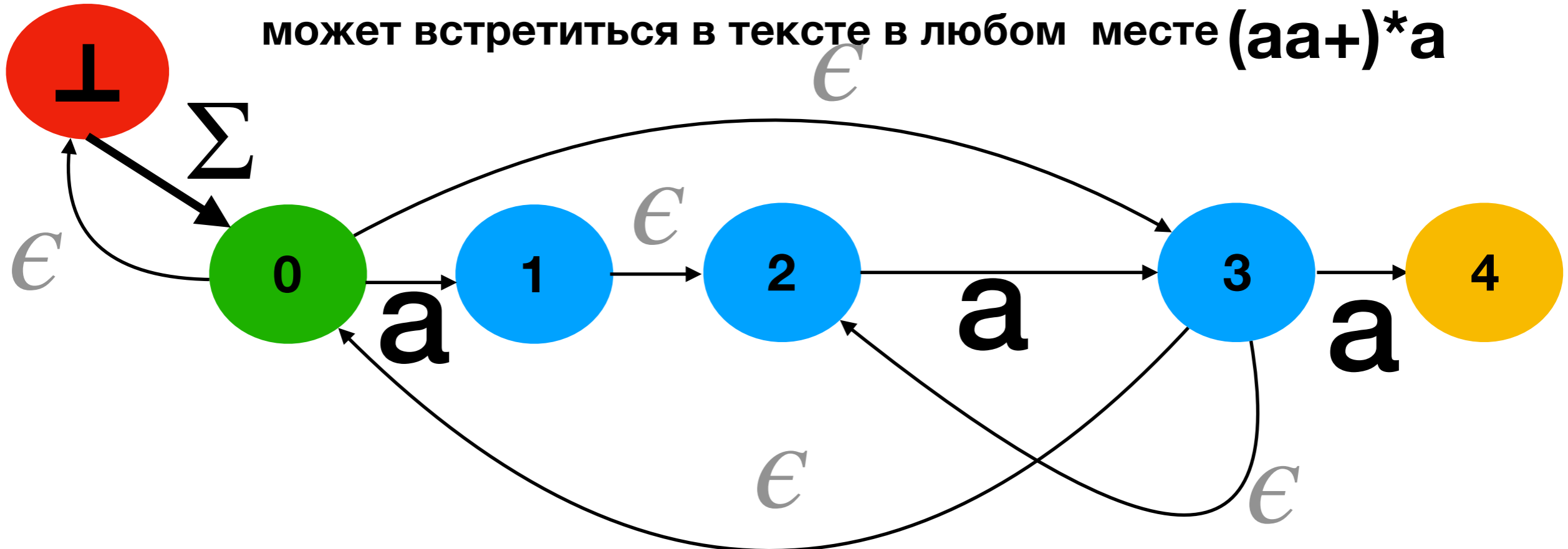
Символ текста	ϵ	a	ϵ	a	ϵ	a	ϵ
Начинаем в 0	0(“”) 3(“”) 2(“”)	1(“a”) 4(“a”) 3(“a”) 0(“-”)	1(“a”) 2(“a”), 2(“-”) 4(“a”) 3(“a”), 3(“-”) 0(“a”), 0(“-”)	Из двух одинаковых состояний выбираем то, у которого входжение длиннее			
	⊥		⊥				

Рассмотрим и еще более сложный пример, мы предполагаем, что паттерн может встретиться в тексте в любом месте $(aa+)^*a$



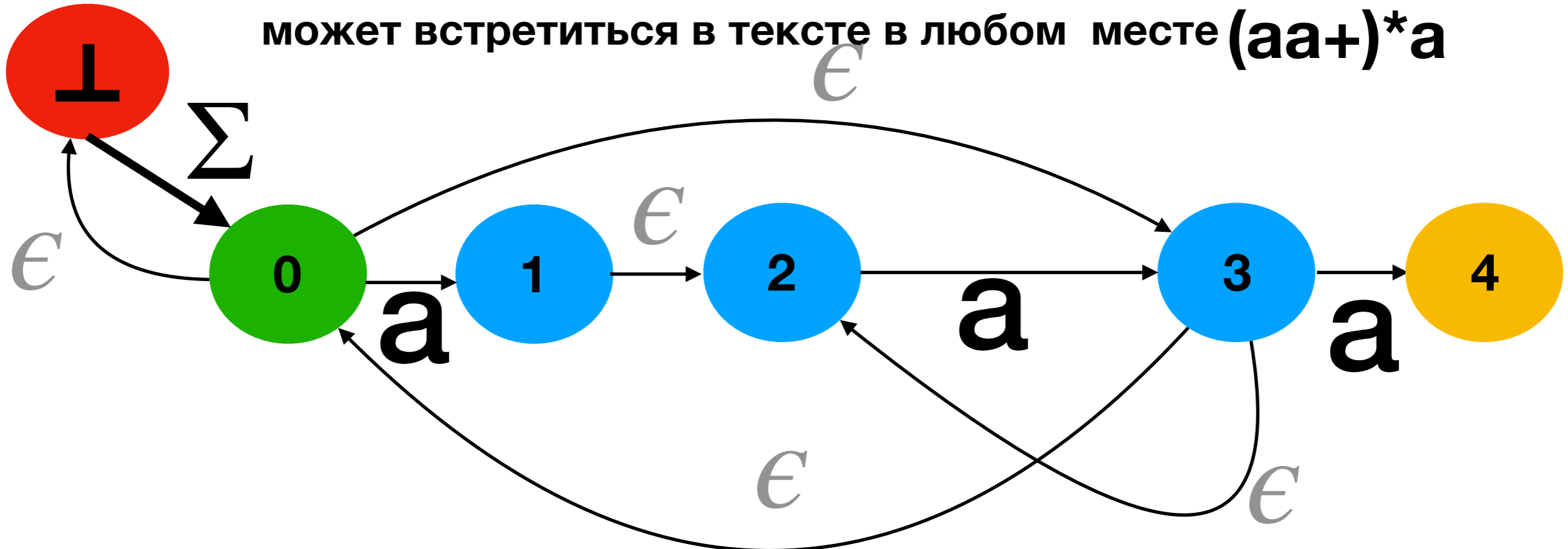
Символ текста	ϵ	a	ϵ	a	ϵ	a	ϵ
Начинаем в 0	0(“”) 3(“”) 2(“”) ⊥	1(“a”) 4(“a”) 3(“a”) 0(“-”)	1(“a”) 2(“a”) 4(“a”) 3(“a”) 0(“a”) ⊥				

Рассмотрим и еще более сложный пример, мы предполагаем, что паттерн может встретиться в тексте в любом месте $(aa+)^*a$



Символ текста	ϵ	a	ϵ	a	ϵ	a	ϵ
Начинаем в 0	0(“”) 3(“”) 2(“”) ┃	1(“a”) 4(“a”) 3(“a”) 0(“-”)	1(“a”) 2(“a”) 4(“a”) 3(“a”) 0(“a”) ┃	3(“aa”) 4(“a”) 4(“aa”) 1(“aa”) 0(“--”)			

Рассмотрим и еще более сложный пример, мы предполагаем, что паттерн может встретиться в тексте в любом месте $(aa+)^*a$



Символ текста	ϵ	a	ϵ	a	ϵ	a	ϵ	
Начинаем в 0	0(“”) 3(“”) 2(“”) └	1(“a”) 4(“a”) 3(“a”) 0(“-”)	1(“a”) 2(“a”) 4(“a”) 3(“a”) 0(“a”) └	3(“aa”) 4(“aa”) 1(“aa”) 0(“--”)	Далее сами			

Сколько искать недетерминированным автоматом в последовательности наиболее длинное/короткое вхождение паттерн?

Используя рассуждения, аналогичные предыдущим, получим, что время поиска наиболее длинного/наиболее короткого вхождения будет занимать.

$$O(M) \cdot N = O(MN)$$

Сколько искать недетерминированным автоматом в последовательности все вхождения паттерна?

Теперь любой переход не в начало автомата создает новое состояние. Мы не можем убирать состояния, как это делали раньше. Пусть в некий момент было M состояний. Каждому из этих состояний соответствует некое число переходов ≥ 1 . Пусть в среднем оно α

Опять итого получим:

$$O(N \cdot M \cdot \alpha^N)$$

Экспонента, все грустно: (- искать все вхождения может оказаться сложно.

Сколько искать недетерминированным автоматом в последовательности все вхождения паттерна?

Теперь любой переход не в начало автомата создает новое состояние. Мы не можем убирать состояния, как это делали раньше. Пусть в некий момент было M состояний. Каждому из этих состояний соответствует некое число переходов ≥ 1 . Пусть в среднем оно α

Опять итого получим:

$$O(N \cdot M \cdot \alpha^N)$$

На самом деле - можно отсеивать полностью одинаковые вхождения.

Тогда:

Сколько искать недетерминированным автоматом в последовательности все вхождения паттерна?

Теперь любой переход не в начало автомата создает новое состояние. Мы не можем убирать состояния, как это делали раньше. Пусть в некий момент было M состояний. Каждому из этих состояний соответствует некое число переходов ≥ 1 . Пусть в среднем оно α

Опять итого получим:

$$O(N \cdot M \cdot \alpha^N)$$

На самом деле - можно отсеивать полностью одинаковые вхождения.

Тогда сложность можно уменьшить. Однако если требовать возможности использования обратных ссылок ($\setminus 1$, вспоминаем Python), то сложность останется экспоненциальной (почему регулярные выражения в Python и в Perl работают экспоненциальное время, причем, даже если вы эти ссылки не используете)

**Пример экспоненциального
паттерна (взрывающийся паттерн)**

$(a^*)_+$