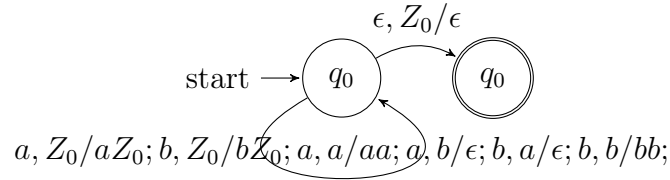


# Домашнее задание №8

Томинин Ярослав, 778 группа

12 ноября 2018 г.

1.



Докажем, почему этот автомат порождает требуемый язык. Назовем язык автомата  $L(A)$

Докажем по индукции по длине слова, что слово  $w \in L(A)$  тогда и только тогда, когда количество букв а и б одинаково.

**База:**  $n=0$ . Существует только одно слово  $\epsilon$ . Правда.

**Переход:** для длины  $n$  доказали, тогда рассмотрим слово длины  $n+1$ , допустим, что оно содержит одинаковое количество букв а и б, но не принимается автоматом. Тогда поймем, что в этом слове есть переход от а к б или ба. Обозначим его за  $\alpha$ . Тогда  $w = w_1\alpha w_2$ . Моймем, что по нашему предположению слово  $w_1w_2$  принимается автоматом, так же заметим, что одно из правил, в которых в стек кладется  $\epsilon$  будет использовано хотя бы 1 раз. (иначе у нас не было бы  $\alpha$ ) Поэтому мы можем рассмотреть слово, в котором нет символов, которые были убраны из стека правилами, кладущими в стек только  $\epsilon$ . Но так как мы убрали символы аб или ба, то разность количества букв а и б по прежнему равна 0, по предположению индукции это слово принимается. Это значит, что и то слово тоже принимается. Теперь докажем, что все слова, принимаемые автоматом, длины  $n+1$  имеют одинаковое количество букв а и б. Наше слово может оказаться в допустимом состоянии только не эпсилон переходу, так как из нашего стека убираются символы аб и ба, а добавляются все прочитанные, то если в конце у нас нет символов, то количество букв а и б одинаковое.

Следовательно, этот автомат порождает нужный язык.

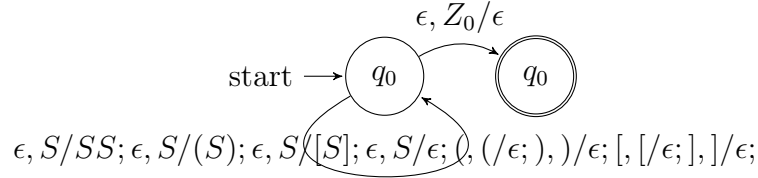
2. Алгоритм подсмотрен из книги.

**Алгоритм построения магазинного автомата по КС-грамматике.:**

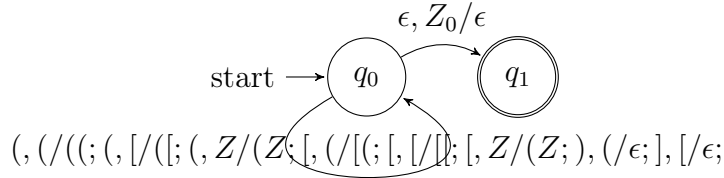
рассмотрим правило вывода  $A \rightarrow \alpha$ . Тогда наш автомат будет состоять из двух состояний  $q_0, q_f$ . Причем для каждого вывода такого вида мы определим функцию перехода [(важно, что делаем отдельный переход по  $\epsilon$  для  $Z$   $\delta(q_0, \epsilon, Z) = (q_0, S)$ )]  $\delta(q_0, \epsilon, Z) = (q_0, \alpha)$ . (В частности для нашей задачи  $\delta(q_0, \epsilon, S) = (q_0, SS), (q_0, (S)), (q_0, [S]), \epsilon$ ). И еще переходы, связанные с обработкой символов: определим переход из начального состояния в начальное, где будем переходить по терминальному символу

к следующим образом:  $\delta(q_0, k, k) = (q_0, \epsilon)$ . Осталось добавить переход в допускающее состояние:  $\delta(\epsilon, Z) = (q_f, \epsilon)$

Построим автомат по этому алгоритму



2)



Докажем следующее утверждение по индукции по длине слова:

**Слово допустимо автоматом тогда и только тогда, когда оно выводимо из грамматики.**

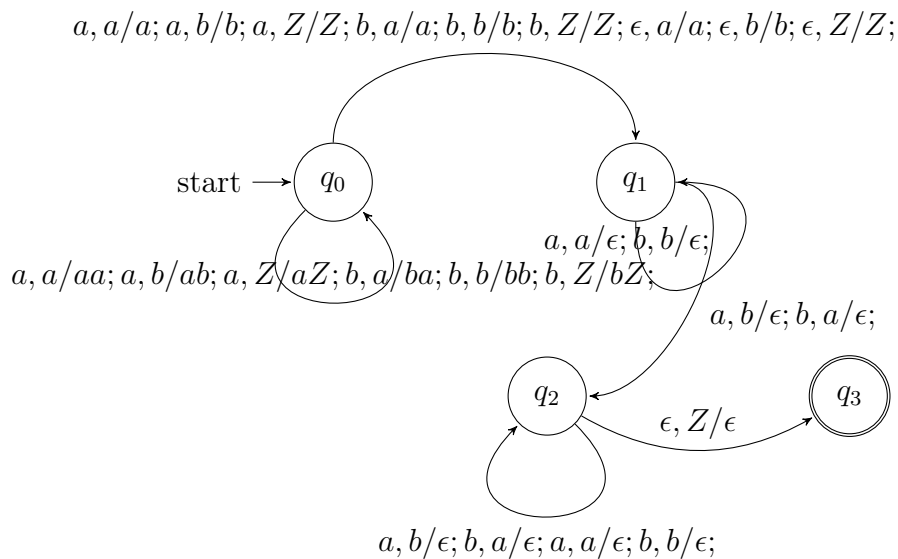
**База:** Рассмотрим  $n=0$ . Из грамматики и из автомата выводимо слово  $\epsilon$ .

**Переход:** предположим, что для слов длины не больше, чем  $n$  выполнено. Предположим, что существует слово, которое является правильной скобочной последовательностью, но при этом не выводимо автоматом. Так как это правильная скобочная последовательность, то существует подслово  $\alpha = ()or[]$ . Заметим, что когда мы доберемся до альфы, то стек не изменит состояния после прочтения альфы (прямо следует из построения). Именно поэтому мы можем рассмотреть слово исключив альфу. Поймем, что это будет правильная скобочная последовательность и по предположению ее должен принимать автомат. Поэтому и наше слово должно приниматься. Противоречие.

Докажем, что все слова, которые выводимы автоматом-правильные скобочные последовательности. Предположим, что слово из неправильной скобочной последовательности выводимо автоматом. Рассмотрим момент, когда мы используем правило удаление скобок из стека (мы его точно используем, так как в конечном итоге в стеке не должно быть скобок). Тогда мы знаем, что эти скобки стояли рядом, так как это правило

было применено впервые. Если мы уберем эти скобки из слова, то состояние автомата не изменится (так как состояние автомата не менялось при прочтении этих скобок). Но по нашему предположению слово не должно допускаться автоматом, так как оно не правильная скобочная последовательность. Противоречие.

3.



#### Доказательство корректности:

**Лемма:** Если наш автомат принимает слово, то при переходе из 0 в 1 он прочитывает ровно половину слова, если это слово четное и половину с округлением вниз, если это слово нечетное.

**Доказательство леммы:** Поймем, что в состояниях 1,2 мы вычеркиваем символы из стека при прочтении символов, поэтому, чтобы мы смогли перейти в допускающее состояние мы должны в порывх вычеркнуть все символы из стека (так как мы вычеркиваем по одному, то количество принятых в состояниях 1, 2 должно быть равно принятых в состоянии 0) и на этот момент (когда мы вычеркнем все символы из стека) все слово должно быть прочитано (иначе получится, что мы умрем в допускающем состоянии). Поэтому лемма доказана.

**Докажем, что палиндромы не принимаются нашим автоматом:** рассмотрим палиндром, по лемме мы перейдем в 1 состояние, когда прочитаем половину. Так как это палиндром, то мы не можем перейти из 1 состояния во второе (так как мы будем читать буквы, которые будут находиться сверху стека, поэтому, когда мы прочитаем все слово, мы будем

находиться в 1 состоянии). Следовательно, палиндром не принимается автоматом.

**Докажем, что не палиндромы принимаются автоматом:** Рассмотрим не палиндром. Тогда, когда мы перейдем в 1 состояние, то мы прочитаем половину слова (понятно, что для нечетного случая мы прочитаем чуть больше половины, но последний прочитанный символ не заносится в стек, поэтому конфигурация автомата при прочтении центральной буквы остается неизменной и мы можем выкинуть центральный символ и свести этот случай к четной длине). Поймем, что у нас всегда будет возможность перейти из 1 состояния во 2 (так как иначе мы уже доказали, что это будет палиндром), поэтому, перейдя во 2 состояние, мы просто будем вычеркивать буквы из стека, пока не дождемся  $Z$ , по которому перейдем в допускающее состояние (на этот момент мы прочитаем все слово, это следует из леммы, так как мы рассматриваем только те случаи, когда мы читаем половину и переходим в 1 состояние, в других автомат умрет).

Мы доказали, что этот автомат принимает требуемый язык.

**4.**

**1)**

Рассмотрим языки

$$L_1 = a^i b^i, i \geq 0;$$

$$L_2 = c^j, j \geq 0;$$

$$L_3 = a^i c^i, i \geq 0;$$

$$L_4 = b, j \geq 0.$$

Тогда язык  $L_5 = \{a^i b^i c^k, i, j \geq 0\}$  равен  $L_5 = L_1 \cdot L_2$

**Запишем грамматику для  $L_1$**

$G_1$ :

$$S_1 \rightarrow aS_1b | \epsilon$$

**Запишем грамматику для  $L_2$**

$G_2$ :

$$S_2 \rightarrow cS_2 | \epsilon$$

Не будем доказывать, что эти грамматики порождают нужные языки, так как это уже было сделано нами не один раз.

Тогда в запишем грамматику объединения языков

**Грамматика для  $L_5$**

$G_5$ :

$$\begin{aligned} S_5 &\rightarrow S_1 S_2 \\ S_1 &\rightarrow a S_1 b | \epsilon \\ S_2 &\rightarrow c S_2 | \epsilon \end{aligned}$$

Этот алгоритм проходилась в классе, поэтому не будем доказывать его корректность.

Построим язык  $L_6$

**Грамматика для  $L_6$**

$G_6$ :

$$\begin{aligned} S_6 &\rightarrow a S_6 c | N \\ N &\rightarrow b N | \epsilon \end{aligned}$$

Пока слово  $\alpha$  содержит  $S$ , то оно стоит по середине слова, так как мы можем применять только 1 правило (здесь пользуемся тем, что у нас будет только один терминал)

Как только мы заменим  $S$ , то количество букв  $a$  и  $c$  будет одинаковым. После замены мы можем добавить произвольное количество букв  $b$ . В конце концов мы можем получить произвольное слово из языка  $L_6$  и только эти слова.

Теперь поймем, что  $L = L_5 \cup L_6$

**Грамматика для  $L$**

$G$ :

$$\begin{aligned} S &\rightarrow S_5 | S_6 \\ S_5 &\rightarrow S_1 S_2 \\ S_1 &\rightarrow a S_1 b | \epsilon \\ S_2 &\rightarrow c S_2 | \epsilon \\ S_6 &\rightarrow a S_6 c | N \\ N &\rightarrow b N | \epsilon \end{aligned}$$

2)

$$\begin{aligned} S &\rightarrow AB|BA|A|B|\epsilon \\ A &\rightarrow aAa|aAb|bAa|bAb|a \\ B &\rightarrow bBb|bBa|aBa|aBb|b \end{aligned}$$

**Любое слово, порождаемое грамматикой не является дополнением тандемных слов:**

Предположим, что наша грамматика допускает тандемное слово. Поймем, что слова четной длины порождаются только через правило АВ, ВА. Следовательно, рассмотрим слово, которое порождается в 1 случае (для второго все аналогично).

$$w = \underbrace{\alpha}_K a \underbrace{\beta}_K \cdot \underbrace{\mu}_M b \underbrace{\delta}_M$$

Заметим, что длина слова  $2K+2M+2$ , но на  $K+1$  и  $2K+M+2$  местах стоят разные буквы. Поэтому мы пришли к противоречию и это было не тандемное слово.

**Наша грамматика порождает любое слово из дополнения языка тандемных слов:**

Рассмотрим слово, если оно нечетной длины, то мы умеем выводить их с помощью нетерминала А (или В, тут без разницы). Рассмотрим слово четной длины  $2n$ . Так как слово не является тандемом, то существует  $i < n$ :  $w[i] \neq w[n+i]$  (иначе бы слово было тандемом). Тогда породим часть  $w[1, 2i-1]$  с помощью А, а часть  $w[2i, 2n]$  с помощью В (если  $w[i] = a$ , иначе первую часть порождаем В, а вторую А). Так как они нечетной длины, то мы можем их породить. Следовательно, любой не тандем порожден нашей грамматикой.

5.

Применим операцию произведения, которую мы делали для регулярных языков со следующими правилами:

- 1) по прежнему состояние автомата соответствует двум состояниям, по одному от каждого автомата.
- 2) за стек будет отвечать только магазинный автомат: то есть переходя из состояния в стек будет ложиться то, что положил бы магазинный автомат, по переходу из состояния, которое соответствует этому состоянию, в состояние, которое соответствует состоянию, в которое мы переходим.
- 3) допускающее состояние - состояние, которое задается двумя допускающими состояниями.
- 4) функция перехода соответствует функции перехода магазинного авто-

мата и автомата, то есть если мы находимся в состоянии  $ij$  и принимаем символ  $k$ , а магазинный автомат переходит из  $i$  по  $k$  в состояние  $q$  и кладет в стек символ  $G$ , а автомат переходит из  $j$  по  $k$  в состояние  $H$ , то наш автомат перейдет в состояние  $qH$  и положит в стек  $G$ .

Теперь поймем, что слово  $w$  будет допускаться полученным автоматом тогда и только тогда, когда слово будет приниматься обоими автоматами, потому что принимающее состояние эквивалентно принимающим состояниям каждого из автоматов. Если же мы попали не в принимающее состояние, то это означает, что слово не принадлежит хотя бы одному из автоматов и поэтому оно не принадлежит нашему автомату.