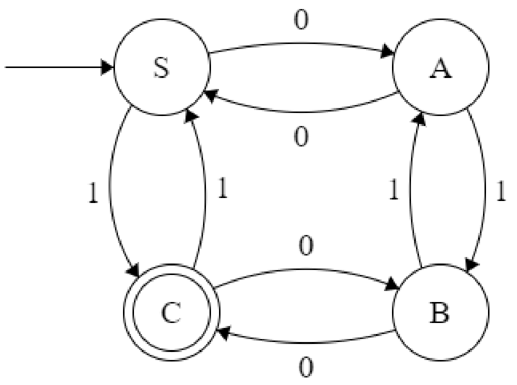
1 Задание 8

1.1 Задача 1

L - язык, состоящий из всех слов в алфавите 0, 1, которые содержат четное число нулей и нечетное число единиц.
1. Построить эквивалентную праволинейную грамматику. Будет ли она одозначной? Построим грамматику по автомату, который принимает этот язык, мы уже строили такой в курсе,



S $\longrightarrow$ 0A | 1C
A $\longrightarrow$ 0S | 1B
B $\longrightarrow$ 0C | 1A
C $\longrightarrow$ 0B | 1S | ε
Эта грамматика будет однозначной, т.к. при каждом шаге слово, которое будет порождаться будет последовательностью каких-то терминальных символов, а на конце слова будет нетерминальный символ. А по каждом нетерминалу переход (правило вывода) определяется однозначно, поэтому для каждого порождаемого слова будет единственный вывод.

2. При переворачивании языка L и получении L^R количество единиц и нулей не изменится в нём, поэтому его будет тот же автомат, что и L. Составим систему уравнений. $S = 0A + 1C$
$A = 0S + 1B$
$B = 0C + 1A$
$C = 0B + 1S + \varepsilon$
Теперь осталось её решить:
$S = 0(0S + 1B) + 1(0B + 1S + \varepsilon)$
$B = 0(0B + 1S + \varepsilon) + 1(0S + 1B)$
$S = 00S + 01B + 10B + 11S + 1$
$B = 00B + 01S + 0 + 10S + 11B$
$B = (00 + 11)B + (01S + 0 + 10S)$

$B = (00 + 11)^*(01S + 0 + 10S)$
$S = 00S + (01 + 10)(00 + 11)^*(01S + 0 + 10S) + 11S + 1$
$S = (00 + (01 + 10)(00 + 11)^*(01 + 10) + 11)S + (01 + 10)(00 + 11)^*0 + 1$
$S = (00 + (01 + 10)(00 + 11)^*(01 + 10) + 11)^*((01 + 10)(00 + 11)^*0 + 1)$
РВ: (00|(01|10)(00|11)*(01|10)|11)*((01|10)(00|11)*0|1)

1.2 Задача 2

Покажем индукцией по длине слова, что любое слово с равным числом букв a и b можно породить этой грамматикой.
Индукция будет по количеству терминальных символов в слове:
База: 2 S $\longrightarrow$ SS $\longrightarrow$ SSS $\longrightarrow$ SbSaS или $\longrightarrow$ SS $\longrightarrow$ SSS $\longrightarrow$ SaSbS. Т.е. любое слово длины 2 можно породить этой грамматикой.
Предположение: для длины 2n
Шаг: для слова длины $2n+2$. Любое слово можно представить в виде $\omega_1 SaSbS\omega_2$ или $\omega_1 SbSaS\omega_2$. ω_1иω_2 выводятся по предположению индукции, $\omega_{1_{[i]}}, \omega_{2_{[j]}} \in \{S, a, b\}$, $i \in \overline{1, |\omega_1|}, j \in \overline{1, |\omega_2|}$. Тогда при выводе получаем, $\omega_1 S\omega_2 \longrightarrow \omega_1 SSS\omega_2 \longrightarrow \omega_1 SaSbS\omega_2$, аналогично и для $\omega_1 SbSaS\omega_2$. То есть мы можем вывести любое слово длины $2n + 2$. Индукция доказана.

1.3 Задача 3

1. Покажите, что язык палиндромов в произвольном авлфавите является КС-языком. Построим КС-грамматику, которая будет задавать язык палиндромов и докажем её корректность. Язык палиндромов = PAL.
Грамматика будет выглядеть вот так: $\forall$ x, y $\in \Sigma^*$ S $\longrightarrow$ xSx | x | ... | ySy | y | ... | ε Покажем, что $L(G) \subseteq PAL$.
Переход S $\longrightarrow$ xSx, на i-м шаге добавляет i-ое место с конца и i-ое место с начала, а переходы S $\longrightarrow$ x влияют только на середину слова. То есть получается палиндром нечетной длины, или не добавляется символ, а добавляется ε и получается палиндром четной длины. Доказано.
PAL $\subseteq L(G)$. Любой палиндром ω нечетной длины k = 2n + 1, n $\geq$ 1, можно получить, последовательно применяя правила: S $\vdash \omega_1 S\omega_1 \vdash \omega_1\omega_2 S\omega_2\omega_1$ ... $\vdash^* \omega_1\omega_2 ... \omega_n S\omega_n ... \omega_2\omega_1 \vdash \omega_1\omega_2 ... \omega_n\omega_{n+1}\omega_n ... \omega_2\omega_1$
Любой палиндром ω четной длины получается таким же применением правил, только в конце вместо $\omega_{n+1}\varepsilon$. S $\vdash \omega_1 S\omega_1 \vdash \omega_1\omega_2 S\omega_2\omega_1$ ... $\vdash^* \omega_1\omega_2 ... \omega_n S\omega_n ... \omega_2\omega_1$ $\vdash \omega_1\omega_2 ... \omega_n\omega_n ... \omega_2\omega_1$ Доказано.
Получается, что L(G) = PAL. Получается, что язык палиндромов является КС-языком.
Доказано.

2. Покажите, что дополнение к языку всех палиндромов тоже является КС-языком.

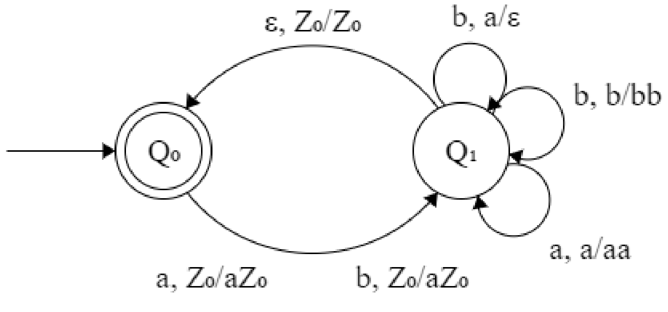
Построим грамматику G, дополнения к языку палиндромов. NEPAL = $\Sigma^* \ PAL$
$\forall x, y \in \Sigma^*, x \neq y$
S $\longrightarrow$ xAy | ... | yAx | ... | xSx | ... | ySy
A $\longrightarrow$ xAy | ... | yAx | ... | xSx | ... | ySy | x | ... | y | ε
L(G) $\subseteq$ NEPAL Пусть ни одно слово ω из L(G) не будет палиндромом. Единственный терминальный символ, из которого можно избавиться от терминального символа будет A, а в A можно попасть только добавив различные символы справа и слева от него. При каждом использовании этого правила мы будем дописывать по символу справа и слева от нетерминала, то есть на k-м шаге получим разные символы на k-м месте сначала и на k-м месте с конца. А чтобы попасть в A, мы хоть раз применим правило S $\longrightarrow$ xAy | yAx , то есть будет такое k, что $\omega_i \neq \omega_{|\omega|+1-i}$, что по определению не будет палиндромом.
NEPAL $\subseteq$ L(G).
Любой палиндром ω длины 2n + 1 или 2n получается последовательным применением правил:

- применяем правило на k-м шаге, если $\omega_k = \omega_{|\omega|+1-k} = x$, S $\longrightarrow$ xSx
- когда на r-м шаге $\omega_r \neq \omega_{|\omega|+1-r}$ применим правило, $x = \omega_r, y = \omega_{|\omega|+1-r}$, S $\longrightarrow$ xSy
- потом пока $l \leq n$ если $\omega_l = \omega_{|\omega|+1-l} = x$, A $\longrightarrow$ xAx, если $\omega_l \neq \omega_{|\omega|+1-l}$, $x = \omega_l, y = \omega_{|\omega|+1-l}$, A $\longrightarrow$ xAy
- для последнего шага, если слово четной длины, то A $\longrightarrow$ ε, если нечетной, то A $\longrightarrow$ ω_{n+1}

Получаем, что L(G) = NEPAL = $\Sigma^* \ PAL$ Получается, что язык дополнения к языку палиндромов является КС-языком.
Доказано.

1.4 Задача 5

Вот детерминированный МП-автомат.



Докажем, что он будет принимать все слова, в которых будет одинаковое количество букв a и b. В стеке будут находится «лишние» символы a, если в считанном слове было больше символов, чем b, точно также и для b. То есть в стеке сможет находится только последовательность букв aa...aaaZ_0, то есть разность между количеством букв a и b в считанном слове, для b аналогично. Получим, что Z_0 будет находится на вершине тогда и только тогда, когда в считанном слове будет одинаковое количество букв a и b, и только тогда будет переход в принимающее состояние Q_0, и переход в Q_1, если остались еще не считанные символы.

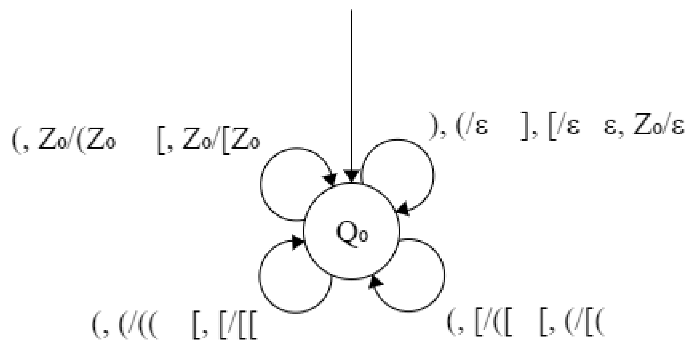
1.5 Задача 6

Язык Дика с двумя типами скобок D_2 порождается грамматикой:

S $\longrightarrow$ SS | (S) | [S] | ε

1. Постройте недетерминированный МП-автомат, распознающий язык D_2.

Построим N-MA автомат $\mathcal{A}$.



Будем доказывает, что этот автомат распознает язык Дика с двумя типами скобок.

$D_2 \subseteq L(\mathcal{A})$. Видно, что любое правильное скобочное выражение будет приниматься $\mathcal{A}$

- скобочные выражения вида [], () - принимаются $\mathcal{A}$: $(Q_0, [], Z_0) \vdash (Q_0,], [Z_0) \vdash (Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$ или же $(Q_0, (), Z_0) \vdash (Q_0,), (Z_0) \vdash (Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$.
- ε - принимается $\mathcal{A}$: $(Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$
- если $x \in L(\mathcal{A}), y \in L(\mathcal{A})$, то $xy \in L(\mathcal{A})$: $(Q_0, xy, Z_0) \vdash^* (Q_0, y, Z_0) \vdash^* (Q_0, \varepsilon, \varepsilon)$
- если $x \in L(\mathcal{A})$, то (x), [x] $\in L(\mathcal{A})$: $(Q_0, (x), Z_0) \vdash (Q_0, x), (Z_0) \vdash^* (Q_0,), (Z_0) \vdash (Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$ или $(Q_0, [x], Z_0) \vdash (Q_0, x], [Z_0) \vdash^* (Q_0,], [Z_0) \vdash (Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$

Теперь докажем, что $L(\mathcal{A}) \subseteq D_2$. Любое слово, которое принимается $\mathcal{A}$ - правильное скобочное выражение.

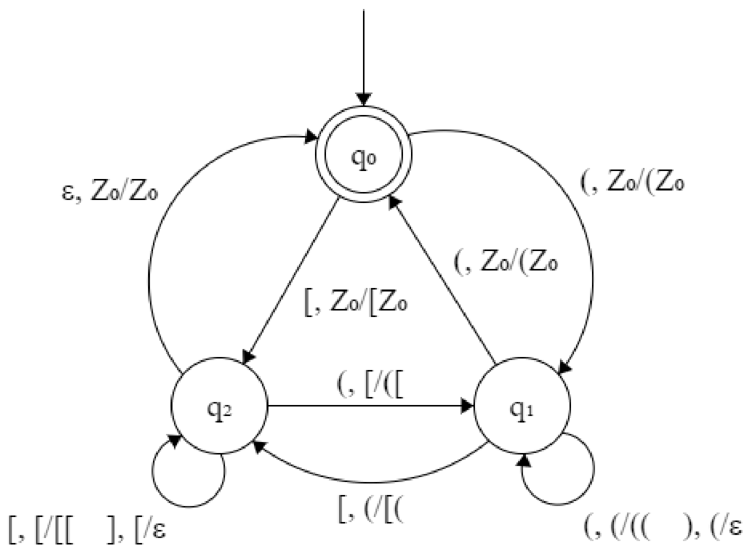
- ε - принимается $\mathcal{A}$: $(Q_0, \varepsilon, Z_0) \vdash (Q_0, \varepsilon, \varepsilon)$ и это правильное скобочное выражение
- У правильных скобочных выражений итоги по [,] и (,) равн нулю, т.к. если открывающих скобок хотя бы одного из видов придет больше, чем закрывающих, т.е. скобочный итог по одному из видом будет положительным, то получится, что стек не будет пуст (приход открывающей скобки увиличивает число символов, которые находятся в стеке).
- Переходы допускают поступление закрывающей скобки какого-либо вида только в случае, если такая же открывающая скобка находится на верхушке стека, т.е. если между закрывающей и открывающей скобкой одинакового вида лежит правильное скобочное выражение.

Получаем, что $L(\mathcal{A}) = D_2$

Доказано

2. Постройте детерминированный МП-автомат, распознающий язык D_2 (достаточно выполнить только этот пункт).

Построим F-MA автомат $\mathcal{A}$:



Докажем корректность по индукции. $D_2 \subseteq L(\mathcal{A}_D)$

База: ε - принимается. Получается, что автомат перейдет в состояние q_0, обработав это слово.

Шаг: пусть слово ω длины n принимается. Все правильные скобочные выражения имеют четную длину, поэтому следующий шаг будет +2. Слово длины n+2 будет иметь один из видов: $\omega()$, $\omega[]$, $()\omega$, $[]\omega$, (ω), $[\omega]$. Первые четыре варианта будут приниматься автомат: (), [], ω переводят автомат в принимающее состояние q_0. Если слово принимается автоматом, то в стеке автомата после приема слова остается только символ Z_0. То есть всё, что положили в стек, досталось из него.

Получаем, что в пятом и шестом случаях мы кладем в стек либо (, либо [. Потом мы кладем и выниаем из него какие-то символы из слова ω, после ω остается только (или [, потом обработаем) или]. Вынимаем символ из стека и можем перейти в принимающее состояние.
Шаг индукции доказан.

Получается, что любое правильное скобочное выражение принимается автоматом.
$L(\mathcal{A}_D) \subseteq D_2$ Видим, что ни одно слово, которое не будет правильным скобочным выражением, автоматом не принимается.