**Лабораторная работа №4**

**4. Устранить из КС-грамматики бесполезные символы и ε–правила**

4.1 Задана КС-грамматика.

4.2. Устранить бесполезные символы, ε-правила.

4.3. Реализация алгоритма.

G = ({b, c}, {S, A, B}, P, S)

P={S→bA, S→bA, A→Ab, A→ε, B →cB, B→ε}

**Вывод цепочек**

S=>bA=>b

S=>bA=>bAb=>bb

S=>bA=>bAb=>bAbb=>bbb

**Удаление непроизводящих символов.**

Вход: G = (T, V, P, S)

Выход: G' = (T, V', P', S)

Шаг 1.

Шаг 2.

Шаг 3.

Шаг 4.

=

P'={S→bA, A→Ab, A→ε, B →cB, B→ε}

G = ({b, c}, {S, A, B}, P', S)

**Удаление недостижимых символов**

Вход: G = (T, V, P, S)

Выход: G' = (T, V', P', S)

Шаг 1.

Шаг 2.

Шаг 3.

=

P'={S→bA, A→Ab, A→ε}

G' = ({b}, {S, A}, P', S)

**Построение множества укорачивающих нетерминалов**

Вход: G = (T, V, P, S)

Выход:

Шаг 1.

Шаг 2.

Шаг 3.

=

**Удаление эпсилон-правил**

Вход: G = (T, V, P, S)

Выход: G' = (T, V', P', S')

Эпсилон-правило: A→ε

Заменим правила так, чтобы получить эквивалентную грамматику. Получим новые правила: S→b и A→b .

P'={S→bA, A→Ab, S→b, A→b}

Граматика G без бесполезных символов и эпсилон-правил:

G'=({b}, {S, A}, { S→bA, A→Ab, S→b, A→b },S)

**Вывод цепочек**

S =>b

S=>bA=> bb

S=>bA=>bAb=> bbb

**Лабораторная работа №5**

**5. Устранить из KС-грамматики цепные правила и устранить левую рекурсию**

5.1. Устранение цепных правил.

* 1. Устранение левой рекурсии.
  2. Реализация алгоритма.

**Удаление цепных правил**

Вход: G = (T, V, P, S)

Выход: G' = (T, V, P', S)

Цепные правила отсутствуют в изначальной грамматике, следовательно:

P' = P;

G'=G;

**Удаление левой рекурсии**

Вход: G = (T, V, P, S)

Выход: G' = (T, V', P', S)

Шаг 1.V={A, S}={}

Шаг 2.i = 1, A→Ab|b

A→b|b A'

A'→b |bA'

V'={S, A, A'}

Больше леворекурсивных правил нет.

G' = (T, {S, A,A'}, { S→bA, S→b, A→b, A→bA', A'→b, A'→bA'}, S)

**Грамматика без леворекурсивных правил и цепных правил.**

G'=({b}, {S, A, A'}, { S→b, S→bA, A→b, A→bA', A'→b, A'→bA'},S)

**Вывод цепочек**

S =>b

S=>bA=> bb

S=>bA=>bbA'=> bbb

**Лабораторная работа №6**

**6. Определить форму КС-грамматики и сделать ее приведение**

6.1. Задана следующая грамматика.

6.2. Эта же грамматика в приведённом виде:

Определить форму грамматики (Грейбаха и т.д.)

G=({b}, {S, A, A'}, { S→b, S→bA, A→b, A→bA', A'→b, A'→bA'},S)

Грамматика в нормальной форме Грейбаха, так как все правила имеют форму A→aγ, где a — терминал, A — нетерминал (возможно, стартовый, γ — строка из не более, чем двух нетерминалов.

Грамматика не в нормальной форме Хомского, так как имеет правила из нетерминалов и терминалов.

Лабораторная

**Лабораторная работа №7**

1. **Спроектировать МП автомат для приведенной КС-грамматики**

**Приведенная грамматика**

G=({b}, {S, A, A'}, { S→b, S→bA, A→b, A→bA', A'→b, A'→bA'},S)

Алгоритм 3.8.

1. Если A → α - правило грамматики G, то (q, ε, A) = (q,α).

2. δ (q, a, a) = {(q, ε)} для всех a ∈ ∑.

**МП-автомат**

МП = ({q}, {b}, {b, S, A, A'}, δ, q, S, {q})

δ:

1. δ(q, ε, S) = {(q, b), (q, bA)}
2. δ(q, ε, A) = {(q, b), (q, bA')}
3. δ(q, ε, A') = {(q, b), (q, bA')}
4. δ(q, b, b) = {(q, ε)}

**Распознавание цепочек:**

(q, b, S) ├1

(q, b, b) ├4

(q, ε, ε)

(q, bb, S) ├1

(q, bb, bA) ├4

(q, bb, A) ├2

(q, bb, bA') ├4

(q, b, A') ├3

(q, b, b) ├4

(q, ε, ε)

(q, bbb, S) ├1

(q, bbb, bA) ├4

(q, bb, A) ├2

(q, bb, bA') ├4

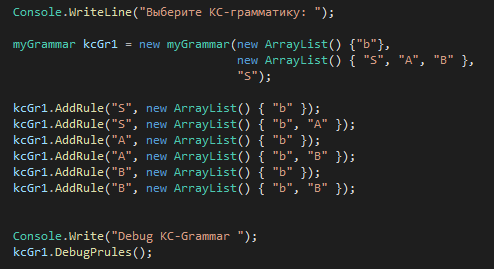
(q, b, A') ├3

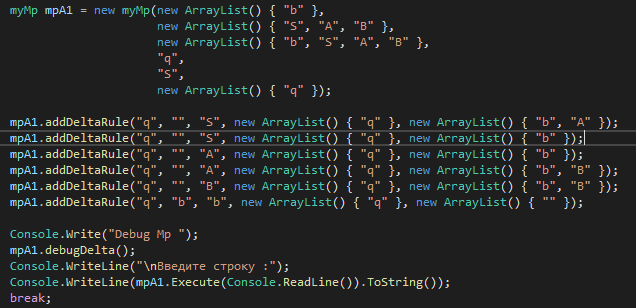
(q, b, b) ├4

(q, ε, ε)

**Лабораторная работа №8**

**8. Реализовать МП автомат для приведенной КС-грамматики**





Prules:

S -> b

S -> bA

A -> b

A -> bB

B -> b

B -> bB

Deltarules :

delta(Q q,T ,Z S) = (Q q,Z bA)

delta(Q q,T ,Z S) = (Q q,Z b)

delta(Q q,T ,Z A) = (Q q,Z b)

delta(Q q,T ,Z A) = (Q q,Z bB)

delta(Q q,T ,Z B) = (Q q,Z bB)

delta(Q q,T b,Z b) = (Q q,Z )

Введите строку :

bb

delta(Q q,T ,Z S) = (Q q,Z bA)

delta(Q q,T b,Z b) = (Q q,Z )

delta(Q q,T ,Z A) = (Q q,Z b)

delta(Q q,T b,Z b) = (Q q,Z )

True