

**Лабораторные работы по курсу
«Системное программное обеспечение»**

1. Спроектировать грамматику по заданному языку L
2. Спроектировать конечный автомат, составить диаграмму переходов КА и реализовать
3. Определить свойства КА. Построить НДКА. Реализовать преобразование НДКА в ДКА.
4. Устранить из КС-грамматики бесполезные символы и ε -правила
5. Устранить из КС-грамматики цепные правила и устранить левую рекурсию
6. Определить форму КС-грамматики и сделать ее приведение
7. Спроектировать МП автомат для приведенной КС-грамматики
8. Реализовать МП автомат для приведенной КС-грамматики
9. Для $LL(1)$ анализатора построить управляющую таблицу M
10. Аналитически написать такты работы $LL(1)$ анализатора для выведенной цепочки.
11. Реализовать управляющую таблицу M для $LL(1)$ анализатора.
12. Построить замыкание множества ситуаций для пополненной $LR(1)$ грамматики.
13. Определить функцию перехода $g(x)$
14. Построить каноническую форму множества ситуаций.
15. Построить управляющую таблицу для функции перехода $g(x)$ и действий $f(u)$.
16. Реализовать $LR(1)$ -анализатор по управляющей таблице (g, f) для $LR(1)$ грамматики.

Студент: Бирюков В. В.
Группа: 08-207

Руководитель: Семёнов А. С.

Оценка:
Дата:

Вариант 1

Лабораторная работа №1

Формулировка задания

Составить примеры цепочек (1) и спроектировать грамматику заданного языка (2)

$$L = \{0\omega_1+(01)^* \mid \omega_1 \in \{0,1\}^*\}$$

$$(1) L = \{0+01, 00+0101, 01+, 000+010101, 001+0101, \dots\}$$

$$(2) L_1 = \{0+01, 00+0101, 01+, 000+010101, 001+0101\}$$

$$L_1 \equiv L_1(G) \subset L$$

Грамматика

$G = (T, V, S_0, P)$, где T – конечное множество терминальных символов, V – конечное множество нетерминальных символов, S_0 – начальное состояние, P – множество правил.

P :

1. $S_0 \rightarrow 0A$
2. $A \rightarrow 0A$
3. $A \rightarrow 1A$
4. $A \rightarrow +B$
5. $A \rightarrow +$
6. $B \rightarrow 0C$
7. $C \rightarrow 1B$
8. $C \rightarrow 1$

Таким образом, грамматика имеет вид:

$$G = (\{0,1,+\}, \{A,B,C\}, 0A, P)$$

Вывод цепочек

$$S_0 \Rightarrow 0A \Rightarrow 0+B \Rightarrow 0+0C \Rightarrow 0+01$$

$$S_0 \Rightarrow 0A \Rightarrow 00A \Rightarrow 00+B \Rightarrow 00+0C \Rightarrow 00+01B \Rightarrow 00+010C \Rightarrow 00+0101$$

$$S_0 \Rightarrow 0A \Rightarrow 01A \Rightarrow 01+$$

$$S_0 \Rightarrow 0A \Rightarrow 00A \Rightarrow 000A \Rightarrow 000+B \Rightarrow 000+0C \Rightarrow 000+01B \Rightarrow 000+010C \Rightarrow 000+0101B \Rightarrow 000+01010C \Rightarrow 000+010101$$

$$S_0 \Rightarrow 0A \Rightarrow 00A \Rightarrow 001A \Rightarrow 001+B \Rightarrow 001+0C \Rightarrow 001+01B \Rightarrow 001+010C \Rightarrow 001+0101$$

Регулярность языка

$$(1) \text{ Для регулярного языка } L = \{0\omega_1+(01)^* \mid \omega_1 \in \{0,1\}^*\} \Rightarrow$$

$$(2) \text{ существует целое } (\exists p=2 \geq 1 \text{ такое что}$$

$$(3) \text{ для всех } (\forall w=01+01 \in L(|w| \geq p) \Rightarrow$$

$$(4) \text{ существует } (\exists x=0, y=1, z=+01 \in \Sigma^* \text{ такое что } (w=xyz \Rightarrow$$

1. $(|y|=|1|\geq 1)$, цикл у должен быть накачан хотя бы длиной 1 и
2. $|xy|=|01|\leq p=2$, цикл должен быть в пределах первых p символов и
3. для всех $i=1\geq 0, (xy^1z=01+01\in L,)))))))$, на x и z ограничений не накладывается.

Лабораторная работа №2

Формулировка задания

Составить на основе разработанной грамматики конечный автомат, распознающий эквивалентный ей язык.

$$L(KA) \equiv L(G)$$

Конечный автомат

KA = (Q, Σ , δ , q_0 , F), где Q – конечное множество состояний, Σ – конечный алфавит входных символов, δ – функция перехода, задаваемая отображением $\delta: Q \times \Sigma \rightarrow Q$, q_0 – начальное состояние автомата, F – множество заключительных состояний.

δ :

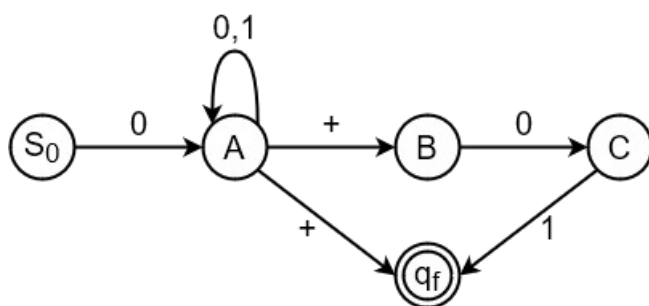
1. $\delta(S_0, 0) = \{A\}$
2. $\delta(A, 0) = \{A\}$
3. $\delta(A, 1) = \{A\}$
4. $\delta(A, +) = \{B, q_f\}$
5. $\delta(B, 0) = \{C\}$
6. $\delta(C, 1) = \{B, q_f\}$

Таким образом конечный автомат имеет вид:

$$KA = (\{S_0, A, B, C, q_f\}, \{0, 1, +\}, \delta, S_0, \{q_f\})$$

KA является недетерминированным.

Диаграмма переходов конечного автомата



Конфигурация КА

$$(S_0, 0+01) \vdash^1 (A, +01) \vdash^4 (B, 01) \vdash^5 (C, 1) \vdash^6 (q_f, \epsilon)$$

$$(S_0, 00+0101) \vdash^1 (A, 0+0101) \vdash^2 (A, +0101) \vdash^4 (B, 0101) \vdash^5 (C, 101) \vdash^6 (B, 01) \vdash^5 (C, 1) \vdash^6 (q_f, \epsilon)$$

$$(S_0, 01+) \vdash^1 (A, 1+) \vdash^3 (A, +) \vdash^6 (q_f, \epsilon)$$

$$\begin{aligned}
& (S_0, 000+010101) \vdash^1 (A, 00+010101) \vdash^2 (A, 0+010101) \vdash^2 (A, +010101) \vdash^4 (B, 010101) \vdash^5 \\
& (C, 10101) \vdash^6 (B, 0101) \vdash^5 (C, 101) \vdash^6 (B, 01) \vdash^5 (C, 1) \vdash^6 (q_f, \varepsilon) \\
& (S_0, 001+0101) \vdash^1 (A, 01+0101) \vdash^2 (A, 1+0101) \vdash^3 (A, +0101) \vdash^4 (B, 0101) \vdash^5 (C, 101) \vdash^6 \\
& (B, 01) \vdash^5 (C, 1) \vdash^6 (q_f, \varepsilon)
\end{aligned}$$

$$L(KA) \equiv L(G)$$

Лабораторная работа №3

Преобразование НДКА в ДКА

НДКА

```
myAutomate ndka = new myAutomate(  
    new ArrayList() { "S0", "A", "B", "C", "qf" },  
    new ArrayList() { "1", "0", "+" },  
    new ArrayList() { "qf" },  
    "S0");  
  
ndka.AddRule("S0", "0", "A");  
ndka.AddRule("A", "0", "A");  
ndka.AddRule("A", "1", "A");  
ndka.AddRule("A", "+", "B");  
ndka.AddRule("B", "0", "C");  
ndka.AddRule("C", "1", "B");  
ndka.AddRule("C", "1", "qf");
```

ДКА

Automate config:

Q: S0 A C Bqf

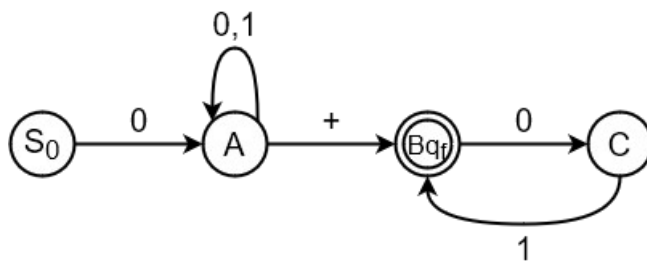
Sigma: 1 0 +

Q0: S0

F: Bqf

DeltaList:

```
: (S0 , 0 ) -> A  
: (A , 1 ) -> A  
: (A , 0 ) -> A  
: (A , + ) -> Bqf  
: (C , 1 ) -> Bqf  
: (Bqf , 0 ) -> C
```



Проверка цепочек

Enter line to execute :

0+01

Length: 4

i :4

curr: Bqf

chineSymbol belongs to language

Enter line to execute :
00+0101
Length: 7
i :7
curr: Bqf
chineSymbol belongs to language

Enter line to execute :
01+
Length: 3
i :3
curr: Bqf
chineSymbol belongs to language

Enter line to execute :
000+010101
Length: 10
i :10
curr: Bqf
chineSymbol belongs to language

Enter line to execute :
001+0101
Length: 8
i :8
curr: Bqf
chineSymbol belongs to language

Лабораторная работа №4

$P = \{S \rightarrow Sb, S \rightarrow F, S \rightarrow cFB, A \rightarrow Ab, A \rightarrow c, B \rightarrow cB, F \rightarrow Ca, C \rightarrow d, C \rightarrow \varepsilon\}$

$G = (\{b, c, a, d\}, \{S, F, B, A, C\}, P, S)$

Вывод цепочек

$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Fbb \Rightarrow Cabb \Rightarrow abb$

$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Sbbb \Rightarrow Fbbb \Rightarrow Cabbb \Rightarrow dabbb$

$S \Rightarrow F \Rightarrow Ca \Rightarrow da$

$S \Rightarrow F \Rightarrow Ca \Rightarrow a$

$S \Rightarrow cFB \Rightarrow cCaB \Rightarrow cdaB \Rightarrow cdacB \Rightarrow cdaccB \Rightarrow cdacccB \Rightarrow cdaccccB$

Алгоритм удаления бесполезных символов

Алгоритм удаления непроезжающих символов

Вход: $G = (T, V, P, S)$

Выход: $G' = (T, V', P', S)$

Шаг 1: $V_p^0 = \emptyset$

Шаг 2: $V_p^1 = \{A, C\}$

Шаг 3: $V_p^2 = \{A, C, F\}$

Шаг 4: $V_p^3 = \{A, C, F, S\}$

Шаг 5: $V_p^4 = \{A, C, F, S\}$

$V_p^3 = V_p^4$

$V' = V \cap V_p = \{A, C, F, S\}$

$P' = \{S \rightarrow Sb, S \rightarrow F, A \rightarrow Ab, A \rightarrow c, F \rightarrow Ca, C \rightarrow d, C \rightarrow \varepsilon\}$

$G = (\{b, c, a, d\}, \{S, F, A, C\}, P', S)$

Алгоритм удаления недостижимых символов

Вход: $G = (T, V, P, S)$

Выход: $G = (T', V', P', S)$

Шаг 1: $VT_r^0 = \{S\}$

Шаг 2: $VT_r^1 = \{S, b, F\}$

Шаг 3: $VT_r^2 = \{S, b, F, C, a\}$

Шаг 4: $VT_r^3 = \{S, b, F, C, a, d\}$

Шаг 5: $VT_r^4 = \{S, b, F, C, a, d\}$

$VT_r^3 = VT_r^4$

$VT_r = \{S, b, F, C, a, d\}$

$T' = T \cap VT_r = \{a, b, d\}$

$V' = V \cap VT_r = \{S, F, C\}$

$$P' = \{S \rightarrow Sb, S \rightarrow F, F \rightarrow Ca, C \rightarrow d, C \rightarrow \varepsilon\}$$

$$G = (\{a, b, d\}, \{S, F, C\}, P', S)$$

Алгоритм построение множества укорачивающих нетерминалов

Вход: $G = (T, V, P, S)$

Выход: $V_\varepsilon = \{C\}$

Шаг 1: $V_\varepsilon^0 = \emptyset$

Шаг 2: $V_\varepsilon^1 = \{C\}$

Шаг 3: $V_\varepsilon^2 = \{C\}$

Алгоритм удаления эпсилон-правил

Вход: $G = (T, V, P, S)$

Выход: $G' = (T, V', P', S')$

$$S' = S$$

$$V' = V$$

$$P' = \{S \rightarrow Sb, S \rightarrow F, F \rightarrow Ca, F \rightarrow a, C \rightarrow d\}$$

Грамматика без бесполезных символов и эпсилон-правил

$$G = (\{S, F, C\}, \{a, b, d\}, P, S)$$

$$P = \{S \rightarrow Sb, S \rightarrow F, F \rightarrow Ca, F \rightarrow a, C \rightarrow d\}$$

Вывод цепочек

$$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Fbb \Rightarrow abb$$

$$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Sbbb \Rightarrow Fbbb \Rightarrow Cabbb \Rightarrow dabbb$$

$$S \Rightarrow F \Rightarrow Ca \Rightarrow da$$

$$S \Rightarrow F \Rightarrow a$$

Лабораторная работа №5

$G = (\{S, F, C\}, \{a, b, d\}, P, S)$

$P = \{S \rightarrow Sb, S \rightarrow F, F \rightarrow Ca, F \rightarrow a, C \rightarrow d\}$

Алгоритм удаления цепных правил

Вход: $G = (T, V, P, S)$

Выход: $G' = (T, V, P', S)$

$V_S = \{S, F\}, V_F = \{F\}, V_C = \{C\}$

Шаг 1: $S \rightarrow F$

$S \rightarrow Ca \mid a$

Шаг 2: Цепных правил не осталось

$P' = \{S \rightarrow Sb, S \rightarrow Ca, S \rightarrow a, C \rightarrow d\}$

$G = (\{S, C\}, \{a, b, d\}, P', S)$

Алгоритм удаления левой рекурсии

Вход: $G = (T, V, P, S)$

Выход: $G' = (T, V', P', S)$

Шаг 1: $V = \{S, C\} = \{A_1, A_2, A_3\}$

Шаг 2: $i = 1. S \rightarrow Sb \mid Ca \mid a$

$S \rightarrow Ca \mid a \mid CaS' \mid aS'$

$S' \rightarrow b \mid bS'$

Шаг 3: $i = 2, j = 1$. Подходящих правил нет.

Шаг 4: $i = 2$. Подходящих правил нет.

Шаг 5: $i = 3, j = 1$. Подходящих правил нет.

Шаг 6: $i = 3, j = 2$. Подходящих правил нет.

Шаг 7: $i = 3$. Подходящих правил нет.

$V' = \{S, C, S'\}$

$P' = \{S \rightarrow CaS', S \rightarrow aS', S' \rightarrow bS', S' \rightarrow b, S \rightarrow Ca, S \rightarrow a, C \rightarrow d\}$

$G = (\{S, C, S'\}, \{a, b, d\}, P', S)$

Грамматика без цепных правил и левой рекурсии

$G = (\{S, C, S'\}, \{a, b, d\}, P, S)$

$P = \{S \rightarrow CaS', S \rightarrow aS', S' \rightarrow bS', S' \rightarrow b, S \rightarrow Ca, S \rightarrow a, C \rightarrow d\}$

Вывод цепочек

$S \Rightarrow aS' \Rightarrow abS' \Rightarrow abb$

$S \Rightarrow CaS' \Rightarrow daS' \Rightarrow dabS' \Rightarrow dabbS' \Rightarrow dabbbb$

$S \Rightarrow Ca \Rightarrow da$

$S \Rightarrow a$

Лабораторная работа №6

Исходная грамматика

$G = (\{b, c, a, d\}, \{S, F, B, A, C\}, P, S)$

$P = \{S \rightarrow Sb, S \rightarrow F, S \rightarrow cFB, A \rightarrow Ab, A \rightarrow c, B \rightarrow cB, F \rightarrow Ca, C \rightarrow d, C \rightarrow \varepsilon\}$

Вывод цепочек

$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Fbb \Rightarrow Cabb \Rightarrow abb$

$S \Rightarrow Sb \Rightarrow Sbb \Rightarrow Sbbb \Rightarrow Fbbb \Rightarrow Cabbb \Rightarrow dabbb$

$S \Rightarrow F \Rightarrow Ca \Rightarrow da$

$S \Rightarrow F \Rightarrow Ca \Rightarrow a$

$S \Rightarrow cFB \Rightarrow cCaB \Rightarrow cdaB \Rightarrow cdacB \Rightarrow cdaccB \Rightarrow cdacccB \Rightarrow cdaccccB$

Приведённая грамматика

$G = (\{S, F, C, S'\}, \{a, b, d\}, P, S)$

$P = \{S \rightarrow CaS', S \rightarrow aS', S' \rightarrow bS', S' \rightarrow b, S \rightarrow Ca, S \rightarrow a, C \rightarrow d\}$

Грамматика не в нормальной форме Грейбах, так как имеет правила, начинающиеся с нетерминалов.

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

Вывод цепочек

$S \Rightarrow aS' \Rightarrow abS' \Rightarrow abb$

$S \Rightarrow CaS' \Rightarrow daS' \Rightarrow dabS' \Rightarrow dabbS' \Rightarrow dabbbb$

$S \Rightarrow Ca \Rightarrow da$

$S \Rightarrow a$

Лабораторная работа №7

Приведённая грамматика

$G = (\{S, C, S'\}, \{a, b, d\}, P, S)$

$P = \{S \rightarrow CaS', S \rightarrow aS', S \rightarrow Ca, S \rightarrow a, S' \rightarrow bS', S' \rightarrow b, C \rightarrow d\}$

МП-автомат

$МП = (\{q\}, \{a, b, d\}, \{a, b, d, S, C, S'\}, \delta, q, S, \{q\})$

δ :

1. $\delta(q, \epsilon, S) = \{(q, CaS'), (q, aS'), (q, Ca), (q, a)\}$
2. $\delta(q, \epsilon, S') = \{(q, bS'), (q, b)\}$
3. $\delta(q, \epsilon, C) = \{(q, d)\}$
4. $\delta(q, a, a) = \{(q, \epsilon)\}$
5. $\delta(q, b, b) = \{(q, \epsilon)\}$
6. $\delta(q, d, d) = \{(q, \epsilon)\}$

Распознавание цепочек

1. $(q, abb, S) \vdash^1 (q, abb, aS') \vdash^4 (q, bb, S') \vdash^2 (q, bb, bS') \vdash^5 (q, b, S') \vdash^2 (q, b, b) \vdash^5 (q, \epsilon, \epsilon)$
2. $(q, dabbbb, S) \vdash^1 (q, dabbbb, CaS') \vdash^3 (q, dabbbb, daS') \vdash^6 (q, abbbb, aS') \vdash^4 (q, bbb, S') \vdash^2 (q, bbb, bS') \vdash^5 (q, bb, S') \vdash^2 (q, bb, bS') \vdash^5 (q, b, S') \vdash^2 (q, b, b) \vdash^5 (q, \epsilon, \epsilon)$
3. $(q, da, S) \vdash^1 (q, da, Ca) \vdash^3 (q, da, da) \vdash^6 (q, a, a) \vdash^4 (q, \epsilon, \epsilon)$
4. $(q, a, S) \vdash^1 (q, a, a) \vdash^4 (q, \epsilon, \epsilon)$

Лабораторная работа №8

```
myMp mp = new myMp(new ArrayList() { "a", "b", "d" },
    new ArrayList() { "S", "C", "D" },
    new ArrayList() { "S", "C", "D", "a", "b", "d" },
    "q",
    "S",
    new ArrayList() { "q" });
mp.addDeltaRule("q", "", "S", new ArrayList() { "q" }, new ArrayList() { "a" });
mp.addDeltaRule("q", "", "S", new ArrayList() { "q" }, new ArrayList() { "C", "a",
"D" });
mp.addDeltaRule("q", "", "S", new ArrayList() { "q" }, new ArrayList() { "a", "D" });
mp.addDeltaRule("q", "", "S", new ArrayList() { "q" }, new ArrayList() { "C", "a" });
mp.addDeltaRule("q", "", "D", new ArrayList() { "q" }, new ArrayList() { "b", "D" });
mp.addDeltaRule("q", "", "D", new ArrayList() { "q" }, new ArrayList() { "b" });
mp.addDeltaRule("q", "", "C", new ArrayList() { "q" }, new ArrayList() { "d" });
mp.addDeltaRule("q", "a", "a", new ArrayList() { "q" }, new ArrayList() { "" });
mp.addDeltaRule("q", "b", "b", new ArrayList() { "q" }, new ArrayList() { "" });
mp.addDeltaRule("q", "d", "d", new ArrayList() { "q" }, new ArrayList() { "" });
```

Проверка цепочек

Prules:

```
S -> a
S -> CaD
S -> aD
S -> Ca
D -> bD
D -> b
C -> d
```

Deltarules :

delta(Q q,T ,Z S)	= (Q q,Z CaD)
delta(Q q,T ,Z D)	= (Q q,Z b)
delta(Q q,T ,Z S)	= (Q q,Z a)
delta(Q q,T ,Z S)	= (Q q,Z aD)
delta(Q q,T ,Z S)	= (Q q,Z Ca)
delta(Q q,T ,Z D)	= (Q q,Z bD)
delta(Q q,T ,Z C)	= (Q q,Z d)
delta(Q q,T a,Z a)	= (Q q,Z)
delta(Q q,T b,Z b)	= (Q q,Z)
delta(Q q,T d,Z d)	= (Q q,Z)

Введите строку :

dab

delta(Q q,T ,Z S)	= (Q q,Z CaD)
delta(Q q,T ,Z C)	= (Q q,Z d)
delta(Q q,T d,Z d)	= (Q q,Z)
delta(Q q,T a,Z a)	= (Q q,Z)
delta(Q q,T ,Z D)	= (Q q,Z b)
delta(Q q,T b,Z b)	= (Q q,Z)

True

Лабораторная работа №9

$$G = (V, T, P, S)$$

$$T = \{i, =, *, (,)\}$$

$$V = \{S, F, L\}$$

P:

1. $S \rightarrow F=L$
2. $S \rightarrow L$
3. $F \rightarrow (*L)$
4. $F \rightarrow i$
5. $L \rightarrow F$

Грамматика не принадлежит классу LL(1), так как для правил $S \rightarrow F=L \mid L$:

$$\text{FIRST}(F=L) = \text{FIRST}(L) = \text{FIRST}(F) = \{(, i\}$$

Заметим, что цепное правило $L \rightarrow F$ является единственным правилом для L, следовательно можно заменить L на F в правых частях других правил и удалить L из грамматики. Получим грамматику:

$$G = (\{S, F\}, \{i, =, *, (,)\}, P, S)$$

P:

1. $S \rightarrow F=F$
2. $S \rightarrow F$
3. $F \rightarrow (*F)$
4. $F \rightarrow i$

Проведем левую факторизацию правил $S \rightarrow F=F \mid F$, заменив их на правила $S \rightarrow FS'$, $S' \rightarrow =F \mid \epsilon$. Получим грамматику без неопределённостей:

$$G = (\{S, S', F\}, \{i, =, *, (,)\}, P, S)$$

P:

1. $S \rightarrow FS'$
2. $S' \rightarrow =F$
3. $S' \rightarrow \epsilon$
4. $F \rightarrow (*F)$
5. $F \rightarrow i$

Построение функции FIRST

$$\text{FIRST}(i) = \{i\}, \text{FIRST}(=) = \{=\}, \text{FIRST}(*) = \{*\}, \text{FIRST}() = \{(, \text{FIRST}() = \{\}$$

$$\text{FIRST}(S) = \text{FIRST}(F) = \{(, i\}$$

$$\text{FIRST}(S') = \{=, \epsilon\}$$

$$\text{FIRST}(F) = \{(, i\}$$

$$\text{FIRST}(FS') = \text{FIRST}(F) = \{(, i\}$$

$$\text{FIRST}((\ast F)) = \{\epsilon\}$$

Построение функции FOLLOW

$$\text{FOLLOW}(S) = \{\perp\}$$

$$\text{FOLLOW}(S') = \{\perp\}$$

$$\text{FOLLOW}(F) = \text{FIRST}(S') \setminus \{\epsilon\} \cup \text{FOLLOW}(S') \cup \{\epsilon\} = \{=, \perp, \epsilon\}$$

Построение управляющей таблицы M

M	i	=	*	()	ε
S	FS', 1			FS', 1		
S'		=F, 2				ε, 3
F	i, 5			(F), 4		
i	ВЫПУСК					
=		ВЫПУСК				
*			ВЫПУСК			
(ВЫПУСК		
)					ВЫПУСК	
⊥						ДОПУСК

Лабораторная работа №10

$G = (\{S, S', F\}, \{i, =, *, (,)\}, P, S)$

$P = \{ S \rightarrow FS', S' \rightarrow =F, S' \rightarrow \varepsilon, F \rightarrow (*F), F \rightarrow i \}$

Вывод цепочек

1. $S \Rightarrow^1 FS' \Rightarrow^4 (*F)S' \Rightarrow^4 ((*F))S' \Rightarrow^5 ((*i))S' \Rightarrow^2 ((*i))=F \Rightarrow^4 ((*i))=(*F) \Rightarrow^5 ((*i))=(*i)$
2. $S \Rightarrow S \Rightarrow^1 FS' \Rightarrow^5 iS' \Rightarrow^2 i=F \Rightarrow^5 i=i$
3. $S \Rightarrow^1 FS' \Rightarrow^4 (*F)S' \Rightarrow^4 ((*F))S' \Rightarrow^4 ((*(*F)))S' \Rightarrow^5 ((*(*i)))S' \Rightarrow^6 ((*(*i)))$

Распознавание цепочек

1. $(((*i))=(*i), S^\perp, \varepsilon) \vdash (((*i))=(*i), FS'^\perp, 1) \vdash (((*i))=(*i), (*F)S'^\perp, 14) \vdash (((*i))=(*i), (*F)S'^\perp, 14) \vdash ((i)=(*i), F)S'^\perp, 14) \vdash ((i)=(*i), (*F))S'^\perp, 144) \vdash ((i)=(*i), (*F))S'^\perp, 144) \vdash ((i)=(*i), F))S'^\perp, 144) \vdash ((i)=(*i), i))S'^\perp, 1445) \vdash ((i)=(*i),))S'^\perp, 1445) \vdash ((i)=(*i),))S'^\perp, 1445) \vdash ((i)=(*i), =F)^\perp, 14452) \vdash ((i), F)^\perp, 14452) \vdash ((i), (*F)^\perp, 144524) \vdash ((i), (*F)^\perp, 144524) \vdash ((i), F)^\perp, 144524) \vdash ((i), i)^\perp, 1445245) \vdash ((i),)^\perp, 1445245) \vdash ((\varepsilon,)^\perp, 1445245)$
2. $(i=i, S^\perp, \varepsilon) \vdash (i=i, FS'^\perp, 1) \vdash (i=i, iS'^\perp, 15) \vdash (i=i, S'^\perp, 15) \vdash (i=i, =F)^\perp, 152) \vdash (i, F)^\perp, 152) \vdash (i, i)^\perp, 1525) \vdash ((\varepsilon,)^\perp, 1525)$
3. $(((*(*i))), S^\perp, \varepsilon) \vdash (((*(*i))), FS'^\perp, 1) \vdash (((*(*i))), (*F)S'^\perp, 14) \vdash (((*(*i))), (*F)S'^\perp, 14) \vdash (((*(*i))), F)S'^\perp, 14) \vdash (((*(*i))), (*F))S'^\perp, 144) \vdash (((*(*i))), (*F))S'^\perp, 144) \vdash (((*(*i))), F))S'^\perp, 144) \vdash (((*(*i))), (*F)))S'^\perp, 1444) \vdash (((*(*i))), (*F)))S'^\perp, 1444) \vdash ((i))), F)))S'^\perp, 1444) \vdash ((i))), i)))S'^\perp, 14445) \vdash ((i))),))S'^\perp, 14445) \vdash ((i))),))S'^\perp, 14445) \vdash ((i))),))S'^\perp, 14445) \vdash ((\varepsilon, S'^\perp, 14445) \vdash ((\varepsilon,)^\perp, 144453)$

Лабораторная работа №11

Грамматика

```
myGrammar grammar = new myGrammar(new ArrayList() { "i", "=", "*", "(", ")", "" },
                                     new ArrayList() { "S", "S'", "F" },
                                     "S");

grammar.AddRule("S", new ArrayList() { "F", "S'" });
grammar.AddRule("S'", new ArrayList() { "=", "F" });
grammar.AddRule("S'", new ArrayList() { "" });
grammar.AddRule("F", new ArrayList() { "(", "*", "F", ")" });
grammar.AddRule("F", new ArrayList() { "i" });
```

Построение управляющей таблицы M

Создадим таблицу. Сначала создадим по столбцу для каждого из этих терминалов:

i, =, *, (,), ,

Также создаем строку для Эпсилон

Рассмотрим нетерминал S

Первый символ правила S -> FS' - (

Это правило заносим в таблицу на пересечении строки нетерминала S и столбца терминала (

Первый символ правила S -> FS' - i

Это правило заносим в таблицу на пересечении строки нетерминала S и столбца терминала i

Рассмотрим нетерминал S'

Первый символ правила S' -> =F - =

Это правило заносим в таблицу на пересечении строки нетерминала S' и столбца терминала =

Рассмотрим нетерминал F

Первый символ правила F -> (*F) - (

Это правило заносим в таблицу на пересечении строки нетерминала F и столбца терминала (

Первый символ правила F -> i - i

Это правило заносим в таблицу на пересечении строки нетерминала F и столбца терминала i

Распознавание цепочек

Введите строку:

(*(i))=(*i)

Успех. Строка соответствует грамматике.

1445245

Введите строку:

i=i

Успех. Строка соответствует грамматике.

1525

Введите строку:

(*(*(i))

Успех. Строка соответствует грамматике.

144453