Московский авиационный институт

(национальный исследовательский институт)

Институт «Компьютерные науки и прикладная математика»

**Лабораторные работы**

**по курсу**

**«Системы программирования»**

**IV семестр**

1. Спроектировать грамматику по паттерн-модели регулярного языка.

2. Преобразовать спроектированную грамматику в конечный автомат, составить диаграмму переходов КА и реализовать.

3. Определить свойства КА. Изучить алгоритм преобразования НДКА в ДКА.

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

5. Устранить из KС-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(k) анализатора построить управляющую таблицу M.

10. Аналитически написать правила вывода для цепочки LL(k) анализатора.

11. Реализовать управляющую таблицу M Для LL(k) анализатора.

12. Построить множество LR(0)-таблиц не содержащих ε-правила.

13. Для LR(k) -грамматики cпроектировать матрицу oblow.

14. Определить функции перехода g(X).

15. Определить функцию переноса-свертки f(u).

16. Для функции перехода g(X) и функции переноса-свертки f(u) спроектировать управляющую таблицу.

*Студент:* Деревянко Е.А.*Группа: М8О-206Б-21*

*Руководитель:* Киндинова В.В.

*Оценка:*

*Дата: 02.06.2023*

**Москва. 2023**

**Практическая работа №1 (1-3 лаб.)**

*Лабораторные работы №1-2*

**Формулировка задания**:

Спроектировать грамматику для трёх заданных паттернов. Составить на основе разработанных регулярных грамматик конечные автоматы, распознающие эквивалентные им языки.

Спроектируем грамматику для заданного языка:

**1. pattern = "192\.168\.1\.\d{1, 3}"**

**Автоматная грамматика:**

L(pattern) - L("192\.168\.1\.\d{1,3}") - {"192.168.1.0" ,. . . , "192.168.1.999"}

G(T, V, P, S0) = G({0,…, 9, .}, {S0, A,…, M}, {p1,…, p16}, S0)

*Правила регулярной грамматики:*

p1: S0 → 1A

p2: A → 9B

p3: B → 2C

p4: C → .D

p5: D → 1E

p6: E → 6F

p7: F → 8G

p8: G → .H

p9: H → 1I

p10: I → .J

p11: J → 0K|1K|…|9K

p12: K → ε

p13: K → 0L|1L|…|9L

p14: L → ε

p15: L → 0M|1M|…|9M

p16: M → ε

*Пример цепочек:*

S0 =>1 1A =>2 19B =>3 192C =>4 192.D =>5 192.1E =>6 192.16F =>7 192.168G =>8 192.168.H =>9 192.168.1I =>10 192.168.1.J =>11 192.168.1.5K =>13 192.168.1.55L =>14 192.168.1.55

**Конечный автомат:**

L(КА) = L(G)

КА = (**Q**, **Σ**, **δ**, **S0**, **F**), где

**Q** = { **S0**, **A,…**, **qf** }, **Σ** = { 0, 1,…, 9, . }, **S0** = **S0**, **F** = **qf**,

**δ** = { 1. δ(**S0**, 1) = {**A**},

2. δ(A, 9) = {B},

3. δ(B, 2) = {C},

4. δ(C, .) = {D},

5. δ(D, 1) = {E},

6. δ(E, 6) = {F},

7. δ(F, 8) = {G},

8. δ(G, .) = {H},

9. δ(H, 1) = {I},

10. δ(I, .) = {J},

11. δ(J, 0) = {K},

…

19. δ(J, 9) = {K},

20. δ(K, ε) = {**qf**}

21. δ(K, 0) = {L}

...

30. δ(K, 9) = {L},

31. δ(L, ε) = {**qf**}

32. δ(L, 0) = {M}

...

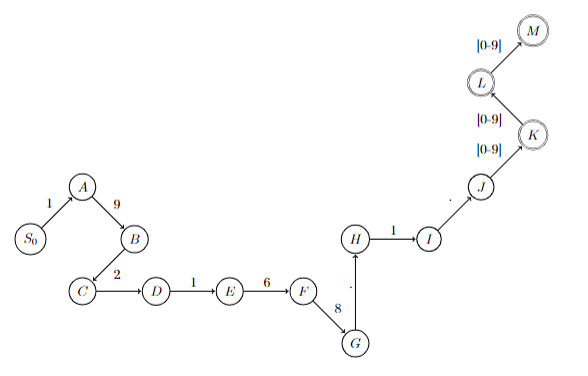
41. δ(L, 9) = {M}

42. δ(M, ε) = {**qf**}

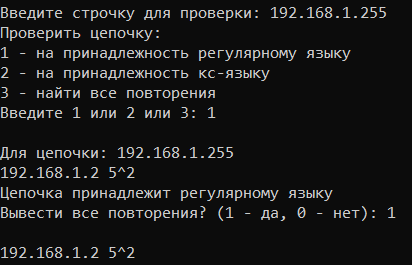
}

*Примеры конфигурации КА:*

1. (S0, 192.168.1.1) ⸠1 (A, 92.168.1.1) ⸠2 (B, 2.168.1.1) ⸠3 (C, .168.1.1) ⸠4 (D,168.1.1) ⸠5 (E, 68.1.1) ⸠6 (F, 8.1.1) ⸠7 (G, .1.1) ⸠8 (H, 1.1) ⸠9 (I, .1) ⸠10 (J,1) ⸠12 (K, ε) ⸠20 (**qf**, ε)



*Лемма о накачке:*



**2. pattern = "(?i)(\W|∧)(baloney|darn|drat|fooey|gosh\sdarnit|heck)(\W|$)"**

**Автоматная грамматика:**

L(pattern) = L("(?i)(\W|∧)(baloney|darn|drat|fooey|gosh\sdarnit|heck)(\W|$)") = {“baloney”, “Baloney”, …}

G(T, V, P, S0) = G({B, b, A, a, L, l, O, o, N, n, E, e, Y, y, D, d, R, r, N, n, T, t, F, f, G, g, S, s, H, h, \s, I, i, C, c, \W}, {**S0**, **A**, ..., **Z**, **AA**, **qf**}, {p1, ..., p34}, **S0**)

*Правила регулярной грамматики:*

p1: S0 → \WS0  
p2: S0 → bA|BA   
p3: S0 → dG|DG   
p4: S0 → gO|GO   
p5: S0 → fL|FL   
p6: S0 → hY|HY   
p7: A → aB|AB   
p8:B → lC|LC   
p9: C → oD|OD   
p10: D → nE|NE   
p11: E → eF|EF   
p12: F → yqf |Yqf

p13: G → aH|AH   
p14: G → rJ|RJ   
p15: H → rI|RI   
p16: I → nqf |Nqf    
p17: J → aK|AK   
p18: K → tqf |Tqf    
p19: L → oM|OM   
p20: M → oE|OE   
p21: O → oP|OP   
p22: P → sQ|SQ   
p23: Q → hR|HR   
p24: R → \sS  
p25: S → dT|DT   
p26: T → aU|AU   
p27: U → rV|RV   
p28: V → nW|NW   
p29: W → iX|IX   
p30: X → tqf |Tqf    
p31: Y → eZ|EZ   
p32: Z → cAA|CAA   
p33: AA → kqf |Kqf    
p34: qf → \Wqf |ε

*Пример цепочек:*

S0 =>1 !S0 =>2 !BA =>7 !BaB =>8 !BaLC =>9 !BaLOD =>10 !BaLOnE =>11 !BaLOnEF =>12 !BaLOnEy

**Конечный автомат:**

L(КА) = L(G)

КА = (**Q**, **Σ**, **δ**, **S0**, **F**), где

**Q** = { **S0**, **A**, ..., Z, AA, **qf** }, **Σ** = {B, b, A, a, L, l, O, o, N, n, E, e, Y, y, D, d, R, r, N, n, T, t, F, f, G, g, S, s, H, h, \s, I, i, C, c, \W }, **S0** = S0, **F**=qf,

**δ** = { 1. δ(**S0**, \W) = {**S0**}, 6. δ(**S0**, g) = {**O**},

2. δ(**S0**, b) = {A},

3. δ(**S0**, B) = {A},

4. δ(**S0**, d) = {G},

5. δ(**S0**, D) = {G},

7. δ(**S0**, G) = {**O**},

8. δ(**S0**, f) = {**L**},

9. δ(**S0**, F) = {**L**},

10. δ(**S0**, h) = {**Y**}

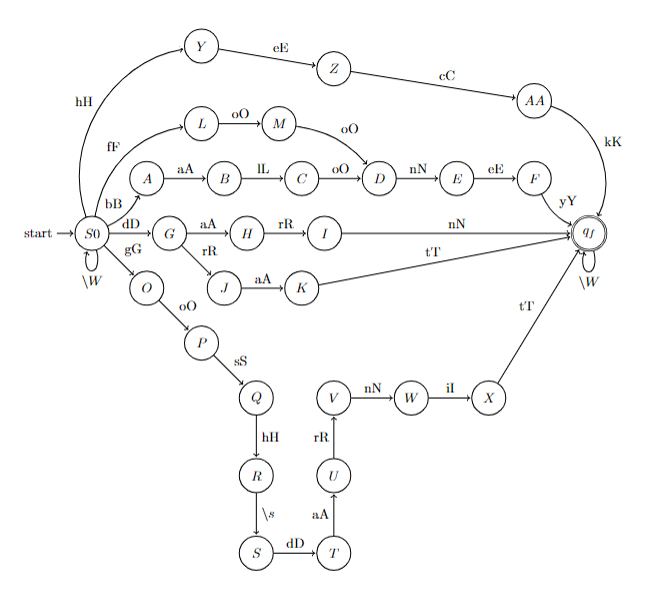
1. δ(**S0**, H) = {**Y**}

…

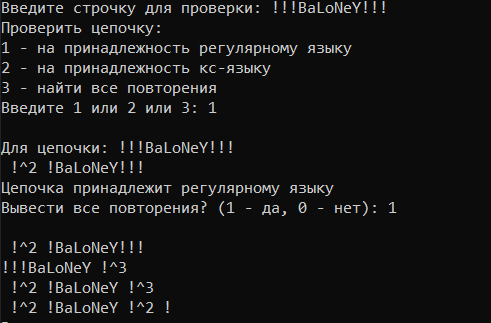
}

*Примеры конфигурации КА:*

(**S0**, !BaLOnEy) ⸠1 (**S0**, BaLOnEy) ⸠3 (**A**, aLOnEy) ⸠13 (**B**, LOnEy) ⸠16 (**C**, OnEy) ⸠18 (**D**, nEy) ⸠19 (**E**, Ey)⸠22 (**F**, y)⸠23 (qf, ε)



*Лемма о накачке:*



*Лабораторная работа №3*

**Формулировка задания:**

Реализовать конечные автоматы, составленные в ЛР №2

**Код программы:**

FSAutomate[] automats = new FSAutomate[] {

new FSAutomate(

new List<Symbol>() { " S01 " , "A1" , "B1" , "C1" , "D1" , "E1" , "F1" , "G1" , "H1" , " I1 " , " J1 " , "K1" , " L1 " , "M1" },

new List<Symbol>() { "0 " , " 1 " , " 2 " , " 3 " , " 4 " , " 5 " , " 6 " , " 7 " , " 8 " , " 9 " , " . " },

new List<Symbol>() { "qf1" },

"S01"

),

new FSAutomate(

new List<Symbol>() {" S02 " , "A2" , "B2" , "C2" , "D2" , "E2" , "F2" , "G2" , "H2" , " I2 " , " J2 " , "K2" , " L2 " , "M2" , "N2" , "O2" , "P2" , "Q2" , "R2" , " S2 " , "T2" , "U2" ,"V2" , "W2" , "X2" , "Y2" , "Z2" , "AA2" , " qf2 " },

new List<Symbol>() { "B" , "b" , "A" , " a " , "L" , " l " , "O" , " o " , "N" , "n" , "E" , " e " , "Y" , " y " , "D" , "d" , "R" , " r " , "N" , "n" , "T" , " t " , "F" , " f " , "G" , " g " , "S" , " s " , "H" , "h" , " " , " I " , " i " , "C" , " c " , @" \W" },

new List<Symbol>() { "qf2" },

"S02"

),

};

//string[] numbers = { " 1 ", " 2 ", " 3 ", " 4 ", " 5 ", " 6 ", " 7 ", " 8 ", " 9 ", " 0 " };

//deltas

automats[0].AddRule("S01","1","A1");

automats[0].AddRule("A1","9","B1");

automats[0].AddRule("B1","2","C1");

automats[0].AddRule("C1",".","D1");

automats[0].AddRule("D1","1","E1");

automats[0].AddRule("E1","6","F1");

automats[0].AddRule("F1", "8", "G1");

automats[0].AddRule("G1", ".", "H1");

automats[0].AddRule("H1", "1", "I1");

automats[0].AddRule("I1", ".", "J1");

automats[0].AddRule("J1", "0", "K1");

automats[0].AddRule("J1", "1", "K1");

automats[0].AddRule("J1", "2", "K1");

automats[0].AddRule("J1", "3", "K1");

automats[0].AddRule("J1", "4", "K1");

automats[0].AddRule("J1", "5", "K1");

automats[0].AddRule("J1", "6", "K1");

automats[0].AddRule("J1", "7", "K1");

automats[0].AddRule("J1", "8", "K1");

automats[0].AddRule("J1", "9", "K1");

automats[0].AddRule("K1", "$", "qf1");

automats[0].AddRule("K1", "0", "L1");

automats[0].AddRule("K1", "1", "L1");

automats[0].AddRule("K1", "2", "L1");

automats[0].AddRule("K1", "3", "L1");

automats[0].AddRule("K1", "4", "L1");

automats[0].AddRule("K1", "5", "L1");

automats[0].AddRule("K1", "6", "L1");

automats[0].AddRule("K1", "7", "L1");

automats[0].AddRule("K1", "8", "L1");

automats[0].AddRule("K1", "9", "L1");

automats[0].AddRule("L1", "$", "qf1");

automats[0].AddRule("L1", "0", "M1");

automats[0].AddRule("L1", "1", "M1");

automats[0].AddRule("L1", "2", "M1");

automats[0].AddRule("L1", "3", "M1");

automats[0].AddRule("L1", "4", "M1");

automats[0].AddRule("L1", "5", "M1");

automats[0].AddRule("L1", "6", "M1");

automats[0].AddRule("L1", "7", "M1");

automats[0].AddRule("L1", "8", "M1");

automats[0].AddRule("L1", "9", "M1");

automats[0].AddRule("M1", "$", "qf1");

automats[1].AddRule("S02", @"\W", "S02");

automats[1].AddRule("S02", "b", "A2");

automats[1].AddRule("S02", "B", "A2");

automats[1].AddRule("A2", "a", "B2");

automats[1].AddRule("A2", "A", "B2");

automats[1].AddRule("B2", "l", "C2");

automats[1].AddRule("B2", "L", "C2");

automats[1].AddRule("C2", "o", "D2");

automats[1].AddRule("C2", "O", "D2");

automats[1].AddRule("D2", "n", "E2");

automats[1].AddRule("D2", "N", "E2");

automats[1].AddRule("E2", "e", "F2");

automats[1].AddRule("E2", "E", "F2");

automats[1].AddRule("F2", "y", "qf2");

automats[1].AddRule("F2", "Y", "qf2");

automats[1].AddRule("S02", "d", "G2");

automats[1].AddRule("S02", "D", "G2");

automats[1].AddRule("G2", "a", "H2");

automats[1].AddRule("G2", "A", "H2");

automats[1].AddRule("H2", "r", "I2");

automats[1].AddRule("H2", "R", "I2");

automats[1].AddRule("I2", "n", "qf2");

automats[1].AddRule("I2", "N", "qf2");

automats[1].AddRule("G2", "r", "J2");

automats[1].AddRule("G2", "R", "J2");

automats[1].AddRule("J2", "a", "K2");

automats[1].AddRule("J2", "A", "K2");

automats[1].AddRule("K2", "t", "qf2");

automats[1].AddRule("K2", "T", "qf2");

automats[1].AddRule("S02", "f", "L2");

automats[1].AddRule("S02", "F", "L2");

automats[1].AddRule("L2", "o", "M2");

automats[1].AddRule("L2", "O", "M2");

automats[1].AddRule("M2", "o", "E2");

automats[1].AddRule("M2", "O", "E2");

automats[1].AddRule("S02", "g", "O2");

automats[1].AddRule("S02", "G", "O2");

automats[1].AddRule("O2", "o", "P2");

automats[1].AddRule("O2", "O", "P2");

automats[1].AddRule("P2", "s", "Q2");

automats[1].AddRule("P2", "S", "Q2");

automats[1].AddRule("Q2", "h", "R2");

automats[1].AddRule("Q2", "H", "R2");

automats[1].AddRule("R2", " ", "S2");

automats[1].AddRule("S2", "d", "T2");

automats[1].AddRule("S2", "D", "T2");

automats[1].AddRule("T2", "a", "U2");

automats[1].AddRule("T2", "A", "U2");

automats[1].AddRule("U2", "r", "V2");

automats[1].AddRule("U2", "R", "V2");

automats[1].AddRule("V2", "n", "W2");

automats[1].AddRule("V2", "N", "W2");

automats[1].AddRule("W2", "i", "X2");

automats[1].AddRule("W2", "I", "X2");

automats[1].AddRule("X2", "t", "qf2");

automats[1].AddRule("X2", "T", "qf2");

automats[1].AddRule("S02", "h", "Y2");

automats[1].AddRule("S02", "H", "Y2");

automats[1].AddRule("Y2", "e", "Z2");

automats[1].AddRule("Y2", "E", "Z2");

automats[1].AddRule("Z2", "c", "AA2");

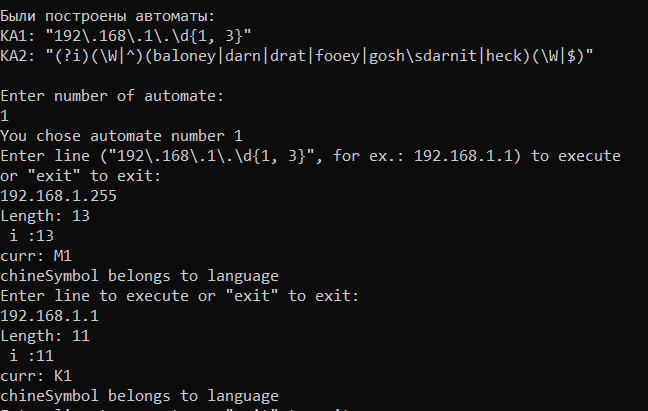
automats[1].AddRule("Z2", "C", "AA2");

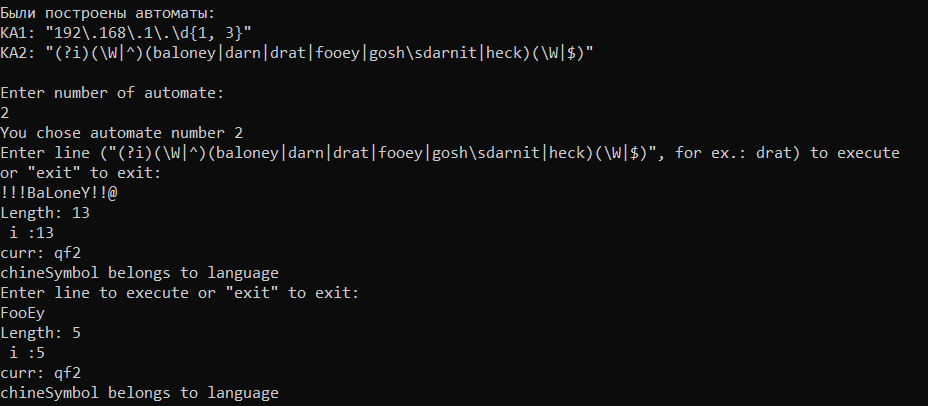
automats[1].AddRule("AA2", "k", "qf2");

automats[1].AddRule("AA2", "K", "qf2");

automats[1].AddRule("qf2", @"\W", "qf2");

**Пример работы программы:**





**Практическая работа №2 (лабораторные 4-8)**

*Лабораторная работа №4:*

**Формулировка задания:**

Устранить из КС-грамматики бесполезные символы и ε–правила

Исходная грамматика:

**G** = (**T**, **V**, **P**, **S0**), где

**T** = {a, d, c, ε, f, p, r, o};

**V** = {S, A, B, C, D, Y, J, H, P, W, O, M};

**P** = {A → CD, C → d, D → c, B → Y, Y → a, S → ABJ, J → ε, H → aPf, W → pOr, S → oM, M → Mc, M → c};

**S0** = S.

p1: A → CD

p2: C → d

p3: D → c

p4: B → Y

p5: Y → a

p6: S → ABJ

p7: J → ε

p8: H → aPf

p9: W → pOr

p10: S → oM

p11: M → Mc

p12: M → c

**Устранить из КС-грамматики длинные правила.**

S → ABJ

S → AE

H → aPf

H → aF

F → Pf

W → pOr

W → pG

G → Or

E → BJ

**Устранить правила с ε**

X = {J}

E → B|BJ

**Устранить цепные цепочек**

**(B;B) (Y;Y) => B** → Y (B;Y) => B → a

**Устранить прямую левую рекурсию**

**M** → Mc

M → c

Замена:

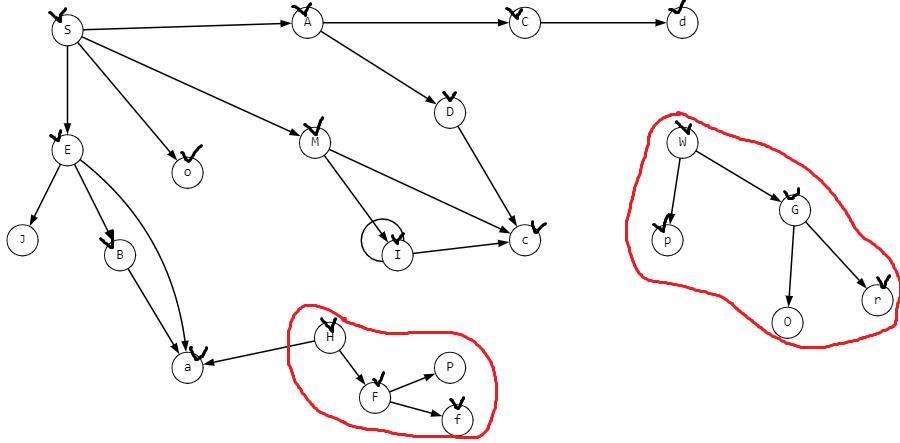
I → cI

I → c

M → cI

M → c

**Устранить бесполезные символы**

****

**Итог:**

p1: S → AE

p2: S → oM

p3: A → CD

p4: C → d

p5: D → c

P6: E → a

P7: M → cI

P8: M → c

P9: I → cI

p10: I → c

*Лабораторная работа №6*

**Формулировка задания:**

Определить форму КС-грамматики и сделать ее приведение

КС-грамматика в нормальной форме Хомского.

Приведённый вид грамматики:

Так, Gпр = ({a, c, d, o}, {S, A, C, D, I, K, L, M, E}, {S → AE, S → KM, K → o, A→CD, C→d, D → c, E → a, M → LI, L → c, M → c, I → LI, I → c}, S)

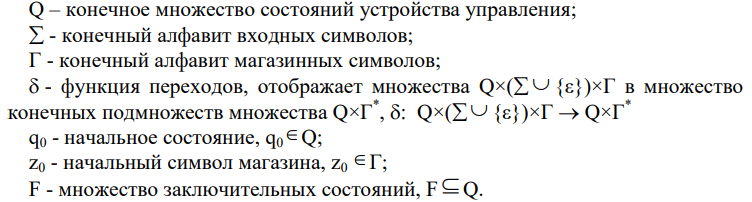
*Лабораторная работа №7*

**Формулировка задания:**

Спроектировать МП-автомат для приведённой КС-грамматики.

*Автоматы с магазинной памятью* (*МП-автоматы*) представляют собой модель распознавателей для языков, задаваемых КС-грамматиками. МП-автоматы имеют вспомогательную память, называемую магазином. В магазин можно поместить неограниченное количество символов. В каждый момент времени доступен только верхний символ магазина. Верхний символом магазина будем считать самый левый символ цепочки.

Так, МП = (Q, Σ, Г, δ, q0, z0, F), где



Приведённая грамматика:

G = (**T**, **V**, **P**, **S**), где

**T** = {a, c, d, o}, **V** = {S, A, C, D, I, K, L, M, E}, **S0** = S

**P**:

p1: S → AE

p2: S → KM

p3: K → o

p4: A → CD

p5: C → d

P6: D → c

P7: E → a

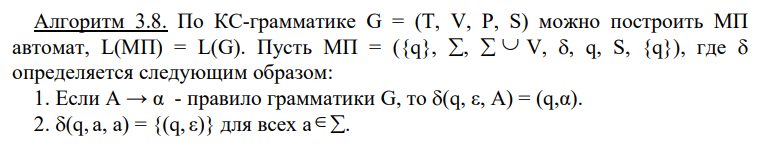
P8: M → LI

P9: L → c

p10: M → c

p11: I → LI

p12: I → c



L(МП) = L(G)

MП = (**Q**, **Σ**, **Г**, **δ**, **q0**, **z0**, **F**):

**Q** = {q}, **Σ** = **T**, **Г** = **T** U **V**, **δ** = **δ**, **q0** = **q0**, **z0** = **S0**, **F** = {q}

МП = ({q}, {a, c, d, o}, {S, A, C, D, I, K, L, M, E}, **δ**, q0, S, {q})

**Определение:** Конфигурацией МП-автомата называется тройка (q, ω, z) ∈ Q×Σ\*×Г\*, где q – текущее состояние управляющего устройства;

ω – необработанная часть входной цепочки (первый символ цепочки ω

находится под входной головкой; если ω = ε, то считается, что вся входная

цепочка прочитана);

z – содержимое магазина (самый левый символ цепочки z считается верхним

символом магазина; если z = ε, то магазин считается пустым).

Функция перехода **δ** определяется согласно алгоритму:



|  |  |
| --- | --- |
| p1: S → AE  p2: S → KM  p3: K → o  p4: A → CD  p5: C → d  P6: D → c  P7: E → a  P8: M → LI  P9: L → c  p10: M → c  p11: I → LI  p10: I → c | 1: δ(q0, ε, S) = (q, AE)  2: δ(q, ε, S) = (q, KM)  3: δ(q, ε, K) = (q, o)  4: δ(q, ε, A) = (q, CD)  5: δ(q, ε, C) = (q, d)  6: δ(q, ε, D) = (q, c)  7: δ(q, ε, E) = (q, a)  8: δ(q, ε, M) = (q, LI)  9: δ(q, ε, L) = (q, c)   1. δ(q, ε, M) = (q, c) 2. δ(q, ε, I) = (q, LI) 3. δ(q, ε, I) = (q, c)   13: δ(q, a, a) = (q, ε) для всех a ∈ Σ |

Последовательность тактов МП-автомата occc:

(q0, occc, S) ⸠2 (q, occc, KM) ⸠3 (q, ccc, M) ⸠9 (q, ccc, LI) ⸠10 (q, cc, I) ⸠12 (q, cc, LI) ⸠10 (q, c, I) ⸠13 (q, c, c) ⸠16 (q, ε, ε)

МП-автоматы – детерменированные автоматы. МП-автомат относится к нисходящим синтаксическим анализаторам, так как строится сверху вниз.

*Лабораторная работа №8*

**Формулировка задания:**

Реализовать спроектированный МП-автомат для приведённой КС-грамматики.

**Код программы:**

var cfgr = new Grammar(new List<Symbol>() { "a", "c", "d", "o" },

new List<Symbol>() { "S", "A", "C", "D", "I", "K", "L", "M", "E" },

"S");

cfgr.AddRule("S",new List<Symbol>() { "A","E" });

cfgr.AddRule("S", new List<Symbol>() { "K", "M" });

cfgr.AddRule("K",new List<Symbol>() { "o" });

cfgr.AddRule("A",new List<Symbol>() { "C","D" });

cfgr.AddRule("C", new List<Symbol>() { "d" });

cfgr.AddRule("D", new List<Symbol>() { "c" });

cfgr.AddRule("E", new List<Symbol>() { "a" });

cfgr.AddRule("M", new List<Symbol>() { "L", "I" });

cfgr.AddRule("L", new List<Symbol>() { "c" });

cfgr.AddRule("M", new List<Symbol>() { "c" });

cfgr.AddRule("I", new List<Symbol>() { "L", "I" });

cfgr.AddRule("I", new List<Symbol>() { "c" });

Console.Write("Debug KC-Grammar ");

cfgr.DebugPrules();

var pda = new PDA(new List<Symbol>() { "q0", "q1","q2","qf"},

new List<Symbol>() { "a", "c", "d", "o" },

new List<Symbol>() {"z0", "a", "c", "d", "o", "S", "A", "C", "D", "I", "K", "L", "M", "E" },

"q0",

"S",

new List<Symbol>() {"qf"});

pda.addDeltaRule("q0", "ε", "S",new List<Symbol>() { "q1" },new List<Symbol>() { "A","E" });

pda.addDeltaRule("q0", "ε", "S", new List<Symbol>() { "q1" }, new List<Symbol>() { "K", "M" });

pda.addDeltaRule("q", "ε", "K",new List<Symbol>() { "q" },new List<Symbol>() {"o" });

pda.addDeltaRule("q", "ε", "A",new List<Symbol>() { "q" },new List<Symbol>() { "C","D" });

pda.addDeltaRule("q", "ε", "C",new List<Symbol>() { "q" },new List<Symbol>() { "d" });

pda.addDeltaRule("q", "ε", "D",new List<Symbol>() { "q" },new List<Symbol>() { "c" });

pda.addDeltaRule("q", "ε", "E",new List<Symbol>() { "q" },new List<Symbol>() { "a" });

pda.addDeltaRule("q", "ε", "M",new List<Symbol>() { "q" },new List<Symbol>() { "L","I" });

pda.addDeltaRule("q", "ε", "L", new List<Symbol>() { "q" }, new List<Symbol>() { "c" });

pda.addDeltaRule("q", "ε", "M", new List<Symbol>() { "q" }, new List<Symbol>() { "c" });

pda.addDeltaRule("q", "ε", "I", new List<Symbol>() { "q" }, new List<Symbol>() { "L","I" });

pda.addDeltaRule("q", "ε", "I", new List<Symbol>() { "q" }, new List<Symbol>() { "c" });

pda.addDeltaRule("q", "o", "o", new List<Symbol>() { "q" }, new List<Symbol>() { "ε" });

pda.addDeltaRule("q", "d", "d", new List<Symbol>() { "q" }, new List<Symbol>() { "ε" });

pda.addDeltaRule("q", "c", "c", new List<Symbol>() { "q" }, new List<Symbol>() { "ε" });

pda.addDeltaRule("q", "a", "a", new List<Symbol>() { "q" }, new List<Symbol>() { "ε" });

pda.addDeltaRule("q", "ε", "d", new List<Symbol>() { "qf" }, new List<Symbol>() { "ε" });

pda.addDeltaRule("q", "ε", "o", new List<Symbol>() { "qf" }, new List<Symbol>() { "ε" });

**Пример работы программы:**

S -> AE

S -> KM

K -> o

A -> CD

C -> d

D -> c

E -> a

M -> LI

L -> c

M -> c

I -> LI

I -> c

Delta rules:

delta(q0,ε,S) -> (q1,A,E)

delta(q0,ε,S) -> (q1,K,M)

delta(q,ε,K) -> (q,o)

delta(q,ε,A) -> (q,C,D)

delta(q,ε,C) -> (q,d)

delta(q,ε,D) -> (q,c)

delta(q,ε,E) -> (q,a)

delta(q,ε,M) -> (q,L,I)

delta(q,ε,L) -> (q,c)

delta(q,ε,M) -> (q,c)

delta(q,ε,I) -> (q,L,I)

delta(q,ε,I) -> (q,c)

delta(q,o,o) -> (q,ε)

delta(q,d,d) -> (q,ε)

delta(q,c,c) -> (q,ε)

delta(q,a,a) -> (q,ε)

delta(q,ε,d) -> (qf,ε)

delta(q,ε,o) -> (qf,ε)

Введите строку, пример :

dca

step 1

delta(q0,ε,S) -> (q1,A,E)

step 1

delta(q,ε,A) -> (q,C,D)

step 1

delta(q,ε,C) -> (q,d)

step 1

delta(q,d,d) -> (q,ε)

step 2 d d

step 3 d

step 1

delta(q,ε,D) -> (q,c)

step 1

delta(q,c,c) -> (q,ε)

step 2 c c

step 3 c

step 1

delta(q,ε,E) -> (q,a)

step 1

delta(q,a,a) -> (q,ε)

step 2 a a

step 3 a

True

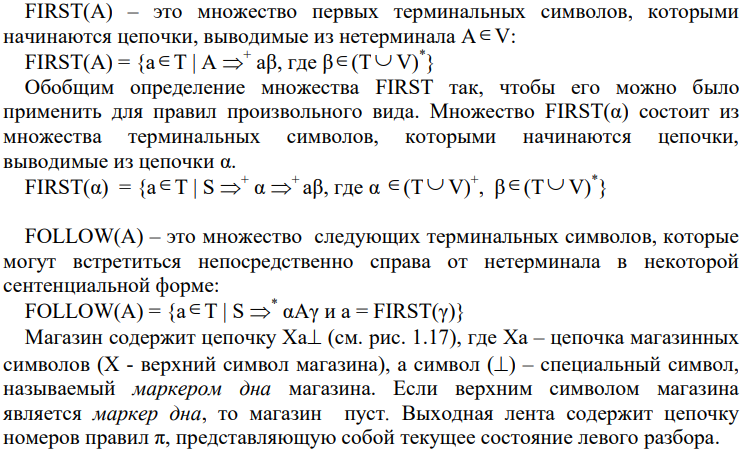
**Практическая работа №3 (лабораторные 9-11)**

*Лабораторная работа №9:*

**Формулировка задания:**

Для LL(k) анализатора построить управляющую таблицу M.

**Определение:** КС-грамматика G = (T, V, P, S) без ε-правил называется простой LL(1) грамматикой (s-грамматикой, разделенной грамматикой), если для каждого v ∈ V все его альтернативы начинаются различными терминальными символами. Единица в названии алгоритма означает, что при чтении анализируемой цепочки, находящейся на входной ленте, входная головка может заглядывать вперед на один символ.

****

Исходная грамматика:

G = (**T**, **V**, **P**, **S**), где

**T** = {a, c, d, o}, **V** = {S, A, C, D, I, K, L, M, E}, **S0** = S

**P**:

p1: S → AE

p2: S → KM

p3: K → o

p4: A → CD

p5: C → d

P6: D → c

P7: E → a

P8: M → LI

P9: L → c

p10: M → c

p11: I → LI

p12: I → c

**Алгоритм построения управляющей таблицы M для LL(1)-грамматики**

*Вход:* LL(1)-грамматика G = (**T**, **V**, **P**, **S**)

*Выход:* Управляющая таблица M для грамматики G.

Таблица M определяется на множестве (**V** U **T** U {Ʇ}) × (**T** U {ε}) по правилам:

1. Если A → β – правило вывода грамматики с номером i, то M(А, a) = (β, i) для всех a ≠ ε, принадлежащих множеству FIRST(β). Если ε ∈FIRST(β), то M(А, b) = (β, i) для всех b ∈ FOLLOW(A).
2. M(a, a) = ВЫБРОС для всех a ∈ **T**.
3. M(Ʇ, ε) = ДОПУСК.
4. В остальных случаях M(X, a) = ОШИБКА для X(**V** U **T** U {Ʇ}) и a ∈ **T** U {ε}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **a** | **c** | **d** | **o** | **ε** |
| **S** |  |  | AE, 1 | KM, 2 |  |
| **A** |  |  | C, 4 |  |  |
| **C** |  |  | ε,5 |  |  |
| **D** |  | ε, 6 |  |  |  |
| **E** | ε, 7 |  |  |  |  |
| **I** |  | L, 11  ε, 12 |  |  |  |
| **K** |  |  |  | ε, 3 |  |
| **L** |  | ε, 9 |  |  |  |
| **M** |  | L, 8 |  |  |  |
| **a** | ВЫБРОС |  |  |  |  |
| **c** |  | ВЫБРОС |  |  |  |
| **d** |  |  | ВЫБРОС |  |  |
| **o** |  |  |  | ВЫБРОС |  |
| **Ʇ** |  |  |  |  | ДОПУСК |
|  |  |  |  |  |  |

Пустые клетки в таблице означают ОШИБКУ.

Аналитичекое представление для таблицы М:

|  |  |
| --- | --- |
| Правило грамматики | Множество |
| p1: S → AE | FIRST(S) = {d} |
| p2: S → KM | FIRST(S) = {o} |
| p3: K → o | FIRST(K) = {o} |
| p4: A → CD | FIRST(A) = {d} |
| p5: C → d | FIRST(C) = {d} |
| p6: D → c | FIRST(D) = {c} |
| p7: E → a | FIRST(E) = {a} |
| p8: M → LI | FIRST(M) = {c} |
| p9: L → c | FIRST(L) = {c} |
| p10: M → c | FIRST(M) = {c} |
| p11: I → LI | FIRST(I) = {c} |
| p12: I → c | FIRST(I) = {c} |

*Лабораторная работа №10:*

**Формулировка задания:**

Аналитически написать правила вывода для цепочки LL(k) анализатора.

|  |  |  |
| --- | --- | --- |
| Правило | Стек | Входящая цепочка |
|  | SꞱ | occc |
| 2 | KMꞱ | occc |
| 3 | oMꞱ | ccc |
| → | MꞱ | ccc |
| 8 | LIꞱ | ccc |
| 9 | cIꞱ | cc |
| → | IꞱ | cc |
| 11 | LIꞱ | cc |
| 9 | cIꞱ | cc |
| → | IꞱ | c |
| 12 | cꞱ | c |
| → | Ʇ | ε |

*Лабораторная работа №11:*

**Формулировка задания:**

Реализовать управляющую таблицу M Для LL(k) анализатора.

**Код программы:**

var LL = new Grammar(new List<Symbol>() { "a","c","d","o" },

new List<Symbol>() { "S","A","C","D","E","I","K","L","M","S0" },

"S0");

LL.AddRule("S0", new List<Symbol>() { "S" });

LL.AddRule("S",new List<Symbol>() { "A","E" });

LL.AddRule("S",new List<Symbol>() { "K","M" });

LL.AddRule("K",new List<Symbol>() { "o" });

LL.AddRule("A",new List<Symbol>() { "C","D" });

LL.AddRule("C", new List<Symbol>() { "d" });

LL.AddRule("D", new List<Symbol>() { "c" });

LL.AddRule("E", new List<Symbol>() { "a" });

LL.AddRule("M", new List<Symbol>() { "L", "I" });

LL.AddRule("L", new List<Symbol>() { "c" });

LL.AddRule("M", new List<Symbol>() { "c" });

LL.AddRule("I", new List<Symbol>() { "L","I" });

LL.AddRule("I", new List<Symbol>() { "c" });

**Результат работы программы:**

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

a, c, d, o,

Также создаем строку для Эпсилон

Рассмотрим нетерминал S

Первый символ правила S -> AE - d

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

Первый символ правила S -> KM - o

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

Рассмотрим нетерминал A

Первый символ правила A -> CD - d

Это правило заносим в таблицу на пересечении строки нетерминала A и столбца терминала d

Рассмотрим нетерминал C

Первый символ правила C -> d - d

Это правило заносим в таблицу на пересечении строки нетерминала C и столбца терминала d

Рассмотрим нетерминал D

Первый символ правила D -> c - c

Это правило заносим в таблицу на пересечении строки нетерминала D и столбца терминала c

Рассмотрим нетерминал E

Первый символ правила E -> a - a

Это правило заносим в таблицу на пересечении строки нетерминала E и столбца терминала a

Рассмотрим нетерминал I

Первый символ правила I -> LI - c

Это правило заносим в таблицу на пересечении строки нетерминала I и столбца терминала c

Первый символ правила I -> c - c

Это правило заносим в таблицу на пересечении строки нетерминала I и столбца терминала c

Рассмотрим нетерминал K

Первый символ правила K -> o - o

Это правило заносим в таблицу на пересечении строки нетерминала K и столбца терминала o

Рассмотрим нетерминал L

Первый символ правила L -> c - c

Это правило заносим в таблицу на пересечении строки нетерминала L и столбца терминала c

Рассмотрим нетерминал M

Первый символ правила M -> LI - c

Это правило заносим в таблицу на пересечении строки нетерминала M и столбца терминала c

Первый символ правила M -> c - c

Это правило заносим в таблицу на пересечении строки нетерминала M и столбца терминала c

Рассмотрим нетерминал S0

Первый символ правила S0 -> S - o

Это правило заносим в таблицу на пересечении строки нетерминала S0 и столбца терминала o

Первый символ правила S0 -> S - d

Это правило заносим в таблицу на пересечении строки нетерминала S0 и столбца терминала d

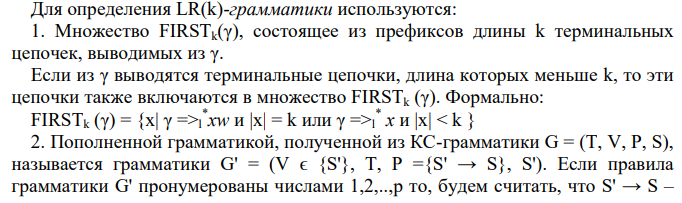
**Практическая работа №4 (лабораторные 12-16)**

**Формулировка задания:**

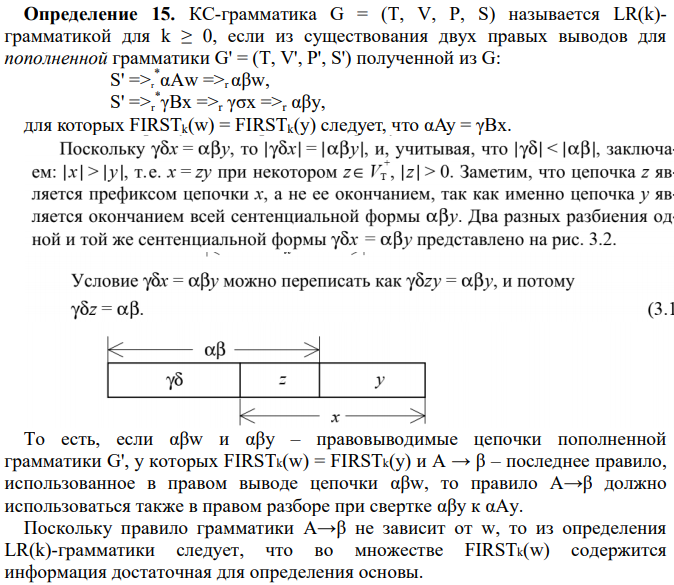
Построить управляющую таблицу M для LR(k)-грамматики, написать правило вывода выделенной строки. Описать работу алгоритма LR(k) анализатора. Построить LR(k) анализатор на основе грамматического вхождения.

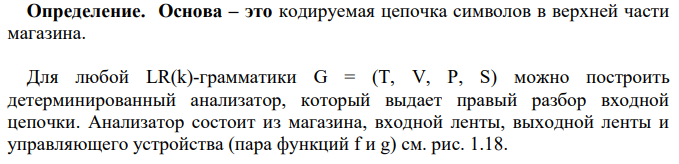
**Синтаксический LR-анализатор** анализирует входную цепочку слева направо (L), и строит правый (R) вывод грамматики.

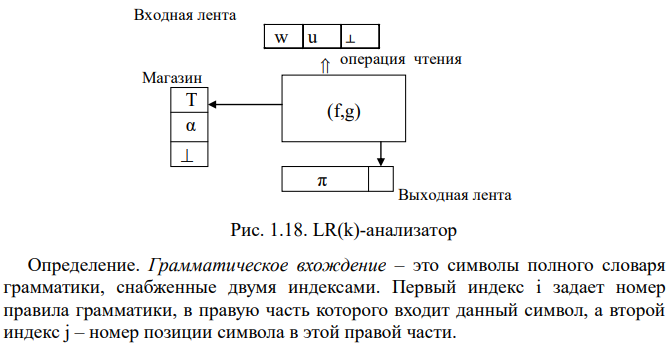
Грамматики, для которых можно построить детерминированный восходящий анализатор, называются LR(k)-*грамматиками* (входная цепочка читается слева (Left) направо, выходом анализатора является правый (Right) разбор, k – число символов входной цепочки, на которое можно “заглянуть” вперёд для выделения основы).











Существует **два способа построения LR(k) анализаторов**:

1. На основе активных префиксов (построения расширенного магазинного алфавита) и отношения OBLOW;
2. На основе LR(0)-ситуаций и функций CLOSURE и GOTO.

Построим двумя способами LR(k) анализатор для заданной грамматики:

G = (**T**, **V**, **P**, **S**), где

**T** = {a, c, d, o}, **V** = {S, A, C, D, I, K, L, M, E}, **S0** = S

**P**:

p1: S → AE

p2: S → KM

p3: K → o

p4: A → CD

p5: C → d

P6: D → c

P7: E → a

P8: M → LI

P9: L → c

p10: M → c

p11: I → LI

p12: I → c

1. **Построение LR(k) анализатора на основе активных префиксов и отношения OBLOW:**

Непосредственно из правил вывода грамматики получим:

A1 OBLOW E1, K2 OBLOW M2, C4 OBLOW D4, L8 OBLOW I8, L11 OBLOW I11

Из определения отношения OBLOW следует, что ⊥ OBLOW Yj ⬄ Yj ∈ OFIRST(S0 = K). Из K можно вывести цепочку K => f1A1. Следовательно, OFIRST(S0 = K) = {a1, S0}, и

⊥ OBLOW S0, ⊥ OBLOW d5, ⊥ OBLOW o3

Рассмотрим правило грамматики с номером (1). Из определения отношения OBLOW следует, что:

f1 OBLOW Yj для всех Yj ∈ OFIRST(A1).

Из A можно вывести цепочки:

A =>2 c2B2 =>3 c2a3C3 =>4 c2a3d4D4 =>5 c2a3d4b5E5 =>6 c2a3d4b5h6F6 =>7 c2a3d4b5h6g7G7 =>8 c2a3d4b5h6g7c8H8 =>9 c2a3d4b5h6g7c8i9

Следовательно, OFIRST(A­1) = {c2} и f1 OBLOW c2

Поступая подобным образом для остальных правил, получим матрицу отношения OBLOW:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **OBLOW** | S | A1 | E1 | K2 | M2 | o3 | C4 | D4 | d5 | c6 | a7 | L8 | I8 | c9 | c10 | L11 | I11 | c12 |
| S |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A1 |  |  | 1 |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |
| E1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K2 |  |  |  |  | 1 |  |  |  |  |  |  | 1 |  | 1 | 1 |  |  |  |
| M2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| o3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C4 |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  |  |  |  |  |  |
| D4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| d5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L8 |  |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 |  | 1 |  | 1 |
| I8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L11 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 | 1 | 1 |
| I11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ⊥ | 1 |  |  |  |  | 1 |  |  | 1 |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | функция переходов g(X) | | | | | | | | | | | | |
| S | A | C | D | I | K | L | M | E | a | c | d | o |
| S0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A1 |  |  |  |  |  |  |  |  | E1 | a7 |  |  |  |
| E1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K2 |  |  |  |  |  |  | L8 | M2 |  |  | c9, c10 |  |  |
| M2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| o3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C4 |  |  |  | D4 |  |  |  |  |  | a7 |  |  |  |
| D4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| d5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c6 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a7 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L8 |  |  |  |  | I8 |  | L11 |  |  |  | c9, c12 |  |  |
| I8 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c9 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c10 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L11 |  |  |  |  | I11 |  | L11 |  |  |  | c9, c12 |  |  |
| I11 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c12 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ⊥ | S0 |  |  |  |  |  |  |  |  |  |  | d5 | o3 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | S | A | C | E | D | I | K | L | M | a | c | d | o |
| {⊥} | {S0} |  |  |  |  |  |  |  |  |  |  | {d5} | {03} |
| {S0} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {D4} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {I8} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {I11} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {L8} |  |  |  |  |  | {I8} |  | {L11} |  |  | {c9,c12} |  |  |
| {L11} |  |  |  |  |  | {I11} |  | {L11} |  |  | {c9,c12} |  |  |
| {M2} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {E1} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {a7} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {c9,c10} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {c9,c12} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {d5} |  |  |  |  |  |  |  |  |  |  |  |  |  |
| {o3} |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| {c9,c10} | {c9,c12} |
| cx | cy |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | S | A | C | E | D | I | K | L | M | a | c | d | o |
| ⊥ | S0 |  |  |  |  |  |  |  |  |  |  | d5 | 03 |
| S0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I8 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I11 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L8 |  |  |  |  |  | I8 |  | L11 |  |  | cy |  |  |
| L11 |  |  |  |  |  | I11 |  | L11 |  |  | cy |  |  |
| M2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a7 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cx |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cy |  |  |  |  |  |  |  |  |  |  |  |  |  |
| d5 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| o3 |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| f(a) | a | c | d | o | ε |
| ⊥ | П | П | П | П |  |
| S0 | С,0 | С,0 | С,0 | С,0 | Д |
| D4 | С,4 | С,4 | С,4 | С,4 | С,4 |
| I8 | С,8 | С,8 | С,8 | С,8 | С,8 |
| I11 | С,11 | С,11 | С,11 | С,11 | С,11 |
| L8 | П | П | П | П |  |
| L11 | П | П | П | П |  |
| M2 | С,2 | С,2 | С,2 | С,2 | С,2 |
| E1 | С,1 | С,1 | С,1 | С,1 | С,1 |
| a7 | С,7 | С,7 | С,7 | С,7 | С,7 |
| cx | П | П | П | П |  |
| cy | С,12 | С,12 | С,12 | С,12 | С,12 |
| d5 | С,5 | С,5 | С,5 | С,5 | С,5 |
| o3 | С,3 | С,3 | С,3 | С,3 | С,3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Символ грамматики | Магазинный символ | Кодируемая цепочка | Операции |
| S00 = S | S0  E1  M2 | ⊥S  AE  KM | Д  П  П |
| A | A1  D4  2 | A  CD | Д  П |
| M | M2  I8  I3 | M  LI | Д  П |
| I | I8  I11 | I  LI | Д  П |
| a | a7  5 | dca | C7 |
| c | c6  c9  c10  c12 | dc  c...  oc  c... | C6  C9  C10  C12 |
| d | d5 | d | C5 |
| o | o3 | o | C3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| f(a) | a | c | d | o | ε |
| ⊥ | П | П | П | П |  |
| S0 | С,0 | С,0 | С,0 | С,0 | Д |
| D4 | С,4 |  |  |  |  |
| I8 |  | С,8 |  |  | С,8 |
| I11 |  |  |  |  | С,11 |
| L8 | П | П | П | П |  |
| L11 | П | П | П | П |  |
| M2 |  |  |  |  | С,2 |
| E1 |  |  |  |  | С,1 |
| a7 |  |  |  |  | С,7 |
| c9 |  | C9 |  |  |  |
| c10 |  |  |  |  | С,10 |
| d5 |  | С,5 |  |  |  |
| o3 |  | С,3 |  |  |  |
| A1 | П |  |  |  |  |
| K2 |  | C2 |  |  |  |
| C4 |  | C4 |  |  |  |
| c6 | C6 |  |  |  |  |
| c12 |  |  |  |  | C12 |