**Лабораторная работа №12-15**

**12. Построить замыкание множества ситуаций для пополненной** LR(1) **грамматики.**

**Формулировка задания**

Определить пополненную LR(1) грамматику.

Определить активные префиксы. Определить функцию перехода g(x). Построить управляющую таблицу для функций перехода и действий.

Определить множество FIRST для LR(1) грамматики. Определить множество LR(0) ситуаций. Построить замыкание CLOSURE(I) множества ситуаций. Определить функцию перехода GOTO(I,х). Построить каноническую форму множества ситуаций. Построить диаграмму переходов автомата. Построить управляющую таблицу для функций перехода и действий.

**Исходная грамматика**

G = (V, T, P, S)

V= {S, F, L}

T = {i, j, & , ^,(,)}

P = {S→ F^L,

S→ (S),

F→ &L,

F→ i,

L→j

}

**Пополненная грамматика**

G = (V, T, P, S)

V= {S, F, L}

T = {i, j, & , ^,(,)}

P = {S'→ S

S→ F^L,

S→ (S),

F→ &L,

F→ i,

L→j

}

**Построение LR(k) анализатора способом использования расширенного магазинного алфавита.**

**Определение активных префиксов**

|  |  |  |
| --- | --- | --- |
| Символ грамматики | Магазинный символ | Кодируемая цепочка |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Построение функции OBLOW**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1 |  |  |  |  |  |  |  |  |
|  |  |  |  | 1 |  |  |  |  |  |  | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 |  |  |  | 1 |  | 1 |  | 1 |  |
|  |  |  |  |  |  |  | 1 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | 1 |  | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 1 |  |  | 1 |  |  | 1 |  | 1 |  |

**Функция переходов g(x)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **g(x)** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Управляющуя таблица**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Функция действий | | | | | | | Функция переходов | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Д |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |
|  | С(1) | С(1) | С(1) | С(1) | С(1) | С(1) | С(1) |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |
|  | С(2) | С(2) | С(2) | С(2) | С(2) | С(2) | С(2) |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |
|  | С(3) | С(3) | С(3) | С(3) | С(3) | С(3) | С(3) |  |  |  |  |  |  |  |  |  |
|  | С(4) | С(4) | С(4) | С(4) | С(4) | С(4) | С(4) |  |  |  |  |  |  |  |  |  |
|  | С(5) | С(5) | С(5) | С(5) | С(5) | С(5) | С(5) |  |  |  |  |  |  |  |  |  |
|  | П | П | П | П | П | П |  |  |  |  |  |  |  |  |  |  |

Д – допуск, С – свёртка, П – перенос, пустые клетки - ОШИБКА

Таблица получилась детерминированной.

Такты работы алгоритма, распознавание цепочек:

**Построение LR(k) анализатора способом грамматических вхождений**

**Построение функции FIRST**

FIRST (S) = {(, &, i}

FIRST (F) = {&, i}

FIRST (L) = {j}

FIRST (F^L) = {&, i}

FIRST ((S)) = {(}

FIRST (&L) = {&}

**Построение функции FOLLOW**

FOLLOW (S) = { ), ┴ }

FOLLOW (F) = { ^ }

FOLLOW (L) = { ^, ┴}

**Замыкание множества ситуаций**

I = CLOSURE({S' → ·S})

Шаг 1. I = CLOSURE({S' → ·S})

Шаг 2. I = {S'→·S*,*S→·F^L*,*S→·(S)}

Шаг 3. I = {S'→·S*,*S→·F^L*,*S→·(S) *,*F→·&L*,*F→·i}

I={S'→·S *,*S→·F^L*,*S→·(S) *,*F→·&L*,*F→·i}

Функция GOTO:

I0=CLOSURE({S'→·S})={S'→·S*,*S→·F^L*,*S→·(S)*,*F→·&L*,*F→·i }

GOTO(I0 *,*S)={S'→S·}=I1

GOTO(I0 *,*F)={S→F·^L}= I2

GOTO(I0 *,*()={S→(·S) *,*S→·F^L*,*S→·(S)*,*F→·&L*,*F→·i }= I3

GOTO(I0 *,*&)={F→&·L*,*L→·j }= I4

GOTO(I0 *,* i)={F→i·}=I5

GOTO(I2 *,* ^)={S→F^·L*,*L→·j }= I6

GOTO(I3 *,*S)={S→(S·)}=I7

GOTO(I3 *,*F)={S→F·^L}= I2

GOTO(I3*,*()={S→(·S) *,*S→·F^L*,*S→·(S)*,*F→·&L*,*F→·i }= I3

GOTO(I3 *,*&)={F→&·L *,*L→·j }= I4

GOTO(I3 *,* i)={F→i·}=I5

GOTO(I4 *,*L)={F→&L·}=I8

GOTO(I4 *,* j)={L→j·}=I9

GOTO(I6 *,*L)={S→F^L·}=I10

GOTO(I6 *,* j)={L→j·}=I9

GOTO(I7 *,* ))={S→(S)·}= I11

**Каноническая форма множества ситуаций:**

C={ I0={S'→·S*,*S→·F^L*,*S→·(S) *,*F→·&L*,*F→·i}*,*

I1={S'→S·}*,*

I2={S→F·^L}*,*

I3={S→(·S)*,*S→·F^L*,*S→·(S) *,*F→·&L*,*F→·i}*,*

I4={F→&·L*,*L→·j }*,*

I5={F→i·}*,*

I6={S→F^·L*,*L→·j }*,*

I7={S→(S·) }*,*

I8={F→&L·}*,*

I9={L→j·}*,*

I10={S→F^L·}*,*

I11={S→(S)·}}

**Диаграмма переходов автомата:**

****

**Управляющая таблица**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
| 0 | П, 5 |  | П, 4 |  | П, 3 |  |  | 1 | 2 |  |
| 1 |  |  |  |  |  |  | Д |  |  |  |
| 2 |  |  |  | П, 6 |  |  |  |  |  |  |
| 3 | П, 5 |  | П, 4 |  | П, 3 |  |  | 7 | 2 |  |
| 4 |  | П, 9 |  |  |  |  |  |  |  | 8 |
| 5 |  |  |  | С, 4 |  |  |  |  |  |  |
| 6 |  | П, 9 |  |  |  |  |  |  |  | 10 |
| 7 |  |  |  |  |  | П, 11 |  |  |  |  |
| 8 |  |  |  | С, 3 |  |  |  |  |  |  |
| 9 |  |  |  | С, 5 |  | С, 5 | С, 5 |  |  |  |
| 10 |  |  |  |  |  | С, 1 | С, 1 |  |  |  |
| 11 |  |  |  |  |  | С, 2 | С, 2 |  |  |  |

Д – допуск, С – свёртка, П – перенос, пустые клетки - ОШИБКА

Такты работы алгоритма, распознавание цепочек:

**16. Реализовать LR(1)-анализатор по управляющей таблице (g,f) для LR(1) грамматики.**

Исходная

КС - грамматика :

Алфавит нетерминальных символов: SFL

Алфавит терминальных символов: : ^()&ij

Правила :

{ S F^L; S (S); F &L; F i; L j }

После удаления е-продукций

Правила:

S F^L

S (S)

F &L

F i

L j

П S

Терминалы : ^()&ij$

Нетерминалы: SFLП

-----

Вычислены множества FIRST для символов грамматики и строк

First( S ): (&i

First( F ): &i

First( L ): j

First( П ): (&i

First( F^L ): &i

First( (S) ): (

First( &L ): &

First( i ): i

First( j ): j

First( S ): (&i

First( F^L ): &i

First( (S) ): (

First( &L ): &

First( i ): i

First( j ): j

First( S ): (&i

CreateCArray:

I0 : { П .S,$; S .F^L,$; S .(S),$; F .&L,^; F .i,^ }

Шаги с GOTO…

Cоздана последовательность С:

I0 { П .S,$; S .F^L,$; S .(S),$; F .&L,^; F .i,^ }

I1 { S (.S),$; S .F^L,); S .(S),); F .&L,^; F .i,^ }

I2 { F &.L,^; L .j,^ }

I3 { F i.,^ }

I4 { L j.,^ }

I5 { П S.,$ }

I6 { S F.^L,$ }

I7 { F &L.,^ }

I8 { S F^.L,$; L .j,$ }

I9 { S (.S),); S .F^L,); S .(S),); F .&L,^; F .i,^ }

I10 { L j.,$ }

I11 { S (S.),$ }

I12 { S F.^L,) }

I13 { S F^L.,$ }

I14 { S F^.L,); L .j,) }

I15 { S (S).,$ }

I16 { L j.,) }

I17 { S (S.),) }

I18 { S F^L.,) }

I19 { S (S).,) }

Создана ACTION таблица

ACTION[0, (] = s 1

ACTION[3, ^] = r F i

ACTION[2, j] = s 4

ACTION[1, (] = s 9

ACTION[0, i] = s 3

ACTION[7, ^] = r F &L

ACTION[1, &] = s 2

ACTION[5, $] = a

ACTION[4, ^] = r L j

ACTION[1, i] = s 3

ACTION[10, $] = r L j

ACTION[9, (] = s 9

ACTION[8, j] = s 10

ACTION[9, i] = s 3

ACTION[14, j] = s 16

ACTION[13, $] = r S F^L

ACTION[12, ^] = s 14

ACTION[18, )] = r S F^L

ACTION[15, $] = r S (S)

ACTION[17, )] = s 19

ACTION[16, )] = r L j

ACTION[6, ^] = s 8

ACTION[9, &] = s 2

ACTION[0, &] = s 2

ACTION[19, )] = r S (S)

ACTION[11, )] = s 15

Создана GOTO таблица

GOTO[0, S] = 5

GOTO[9, S] = 17

GOTO[8, L] = 13

GOTO[14, L] = 18

GOTO[9, F] = 12

GOTO[1, F] = 12

GOTO[2, L] = 7

GOTO[1, S] = 11

GOTO[0, F] = 6

Введите строку:

&j^j

Введена строка: &j^j$

Процесс вывода:

L->jF->&LL->j

S->F^L

Строка допущена

Введите строку:

((i^j))

Введена строка: ((i^j))$

Процесс вывода:

F->i

L->j

S->F^L

S->(S)

S->(S)

Строка допущена

Введите строку:

(&j^j)

Введена строка: (&j^j)$

Процесс вывода:

L->jF->&L

L->j

S->F^L

S->(S)

Строка допущена