

Московский авиационный институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Системы программирования»
Тема: «Построение простого LR-анализатора на основе замыкания
множества ситуаций CLOSURE и функции перехода GOTO»

Студент: Инютин М. А.
Преподаватель: Семёнов А. С.
Группа: М8О-207Б-19
Дата:
Оценка:
Подпись:

Москва, 2021

Задача

Представить алгоритм построения простого LR-анализатора на основе замыкания множества ситуаций CLOSURE и функции перехода GOTO в теоретико-множественном определении. Составить программу на языке C#, реализующую алгоритм простого LR-анализа.

Описание

LR-анализатор принимает решение о выборе «перенос-свёртка» поддерживая состояния, которые отслеживают, где именно в процессе синтаксического анализа мы находимся. Состояния представляют собой множества «ситуаций». LR(0)-ситуация (для краткости – просто ситуация) грамматики G – это продукция G с точкой в некоторой позиции правой части.

Один набор множеств LR(0)-ситуаций, именуемый каноническим набором ϕ , обеспечивает основу для построения детерминированного конечного автомата, который используется для принятия решений в процессе синтаксического анализа. Такой автомат называется LR(0)-автоматом. В частности, каждое состояние LR(0)-автомата представляет множество ситуаций в каноническом наборе ϕ .

Для построения канонического LR(0)-набора мы определяем расширенную грамматику и две функции, CLOSURE и GOTO.

Если I – множество ситуаций грамматики G , то CLOSURE(I) представляет собой множество ситуаций, построенное из I согласно двум правилам:

1. Изначально в CLOSURE(I) добавляются все ситуации из I ;
2. Если $A \rightarrow \alpha B \beta$ входит в CLOSURE(I), а $B \rightarrow \gamma$ является продукцией, то в CLOSURE(I) добавляется ситуация $B \rightarrow \cdot \gamma$, если его там ещё нет. Это правило применяется до тех пор, пока не останутся ситуации, которые могут быть добавлены в CLOSURE(I).

Второй полезной функцией является GOTO(I, X), где I – множество ситуаций, а X – грамматический символ. GOTO(I, X) определяется как замыкание множества всех ситуаций $A \rightarrow \alpha X \beta$ таких, что $A \rightarrow \alpha X \beta$ находится в I . Интуитивно функция GOTO используется для определения переходов в LR(0)-автомате грамматики. Состояние автомата соответствуют множествам ситуаций, и GOTO(I, X) указывает переход из состояния I при входном символе X .

Алгоритм: построение замыкания множества LR(0)-ситуаций $CLOSURE(I)$

Вход: I – множество LR(0)-ситуаций пополненной КС-грамматики G'

Выход: J – замыкание множества ситуаций $CLOSURE(I)$

```
 $J_0 = I$ 
 $J = \emptyset$ 
do {
     $J = J_0$ 
    foreach (  $[A \rightarrow \alpha \cdot B \beta] \in J$  ) {
        foreach (  $B \rightarrow \gamma \in P'$  ) {
            if (  $[B \rightarrow \cdot \gamma] \notin J_0$  ) {
                 $J_0 = J_0 \cup \{[B \rightarrow \cdot \gamma]\}$ 
            }
        }
    }
} while (  $J \neq J_0$  )
```

Пример: пополненная грамматика $G' = (\{i, j, \&, ^, (,)\}, \{S', S, F, L\}, P', S')$

P' :

0. $S' \rightarrow S$
1. $S \rightarrow F \wedge L$
2. $S \rightarrow (S)$
3. $F \rightarrow \&L$
4. $F \rightarrow i$
5. $L \rightarrow j$

$I = \{[S' \rightarrow \cdot S]\}$

$J_0 = \{[S' \rightarrow \cdot S]\}$

Шаг 1.

$J_0 = J_0 \cup \{[S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)]\} = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)]\}$

Шаг 2.

$J_0 = J_0 \cup \{[F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\} = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\}$

Результат:

$J = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\}$

Алгоритм: вычисление множества $GOTO(I, X)$

Вход: I – множество $LR(0)$ -ситуаций пополненной КС-грамматики G' , X – символ КС-грамматики G'

Выход: J – множество ситуаций $GOTO(I, X)$

```
J = ∅
foreach ( [ A → α · X β ] ∈ I ) {
    J = J ∪ CLOSURE( [ [ A → α X · β ] ] )
}
```

Алгоритм: построение канонического набора множеств $LR(0)$ -ситуаций и автомата перехода между ситуациями

Вход: пополненная КС-грамматика $G' = (T, V', P', S')$

Выход: ϕ – канонический набор множеств $LR(0)$ -ситуаций, $LR(0)$ -автомат переходов $KA = (Q = \phi, \Sigma = T \cup V' \cup \{\$, \delta, I_0, q_f)$

```
ϕ₀ = CLOSURE( [ [ S' → · S ] ] )
ϕ = ∅
do {
    ϕ = ϕ₀
    foreach ( I ∈ ϕ ) {
        foreach ( X ∈ Σ ) {
            if ( GOTO(I, X) ≠ ∅ ) {
                if ( GOTO(I, X) ∉ ϕ₀ ) {
                    ϕ₀ = ϕ₀ ∪ GOTO(I, X)
                }
                // Запоминаем все состояния, из которых есть переход
                q_f = q_f ∪ { I }
                // Добавляем новое правило перехода в автомат
                Δ = Δ ∪ { δ(I, X) = { GOTO(I, X) } }
            }
        }
    }
} while ( ϕ ≠ ϕ₀ )
// Вычитаем из всех состояний те, из которых есть переход
q_f = ϕ \ q_f
```

Пример: пополненная грамматика $G' = (\{i, j, \&, ^\wedge, (,)\}, \{S', S, F, L\}, P', S')$

P' :

0. $S' \rightarrow S$
1. $S \rightarrow F \wedge L$
2. $S \rightarrow (S)$
3. $F \rightarrow \&L$
4. $F \rightarrow i$
5. $L \rightarrow j$

$$I_0 = \text{CLOSURE}(\{[S' \rightarrow \cdot S]\}) = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\}$$

$$\phi_0 = \{I_0\}$$

Шаг 1.

$$\text{GOTO}(I_0, S) = \text{CLOSURE}(\{[S' \rightarrow S \cdot]\}) = \{[S' \rightarrow S \cdot]\} = I_1$$

$$\text{GOTO}(I_0, F) = \text{CLOSURE}(\{[S \rightarrow F \cdot \wedge L]\}) = \{[S \rightarrow F \cdot \wedge L]\} = I_2$$

$$\text{GOTO}(I_0, () = \text{CLOSURE}(\{[S' \rightarrow (\cdot S)]\}) = \{[S \rightarrow (\cdot S)], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\} = I_3$$

$$\text{GOTO}(I_0, \&) = \text{CLOSURE}(\{[F \rightarrow \& \cdot L]\}) = \{[F \rightarrow \& \cdot L], [L \rightarrow \cdot j]\} = I_4$$

$$\text{GOTO}(I_0, i) = \text{CLOSURE}(\{[F \rightarrow i \cdot]\}) = \{[F \rightarrow i \cdot]\} = I_5$$

$$\Delta = \Delta \cup \{\delta(I_0, S) = \{I_1\}, \delta(I_0, F) = \{I_2\}, \delta(I_0, () = \{I_3\}, \delta(I_0, \&) = \{I_4\}, \delta(I_0, i) = \{I_5\}\}$$

$$\phi_0 = \phi_0 \cup \{I_1\} \cup \{I_2\} \cup \{I_3\} \cup \{I_4\} \cup \{I_5\} = \{I_0, I_1, I_2, I_3, I_4, I_5\}$$

Шаг 2.

$$\text{GOTO}(I_2, \wedge) = \text{CLOSURE}(\{[S \rightarrow F \wedge \cdot L]\}) = \{[S \rightarrow F \wedge \cdot L], [L \rightarrow \cdot j]\} = I_6, \Delta = \Delta \cup \{\delta(I_2, \wedge) = \{I_6\}\}$$

$$\phi_0 = \phi_0 \cup \{I_6\} = \{I_0, I_1, I_2, I_3, I_4, I_5, I_6\}$$

Шаг 3.

$$\text{GOTO}(I_3, S) = \text{CLOSURE}(\{[S \rightarrow (S \cdot)]\}) = \{[S \rightarrow (S \cdot)]\} = I_7$$

$$\text{GOTO}(I_3, F) = \text{CLOSURE}(\{[S \rightarrow F \cdot \wedge L]\}) = \{[S \rightarrow F \cdot \wedge L]\} = I_2$$

$$\text{GOTO}(I_3, () = \text{CLOSURE}(\{[S \rightarrow (\cdot S)]\}) = \{[S \rightarrow (\cdot S)], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\} = I_3$$

$$\text{GOTO}(I_3, \&) = \text{CLOSURE}(\{[F \rightarrow \& \cdot L]\}) = \{[F \rightarrow \& \cdot L], [L \rightarrow \cdot j]\} = I_4$$

$$\text{GOTO}(I_3, i) = \text{CLOSURE}(\{[F \rightarrow i \cdot]\}) = \{[F \rightarrow i \cdot]\} = I_5$$

$$\Delta = \Delta \cup \{\delta(I_3, S) = \{I_7\}, \delta(I_3, F) = \{I_2\}, \delta(I_3, () = \{I_3\}, \delta(I_3, \&) = \{I_4\}, \delta(I_3, i) = \{I_5\}\}$$

$$\phi_0 = \phi_0 \cup \{I_7\} \cup \{I_2\} \cup \{I_3\} \cup \{I_4\} \cup \{I_5\} = \{I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7\}$$

Шаг 4.

$$\text{GOTO}(I_4, L) = \text{CLOSURE}(\{[F \rightarrow \&L \cdot]\}) = \{[F \rightarrow \&L \cdot]\} = I_8, \Delta = \Delta \cup \{\delta(I_4, L) = \{I_8\}\}$$

$$\text{GOTO}(I_4, j) = \text{CLOSURE}(\{[L \rightarrow j \cdot]\}) = \{[L \rightarrow j \cdot]\} = I_9, \Delta = \Delta \cup \{\delta(I_4, j) = \{I_9\}\}$$

$$\phi_0 = \phi_0 \cup \{I_8\} \cup \{I_9\} = \{I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9\}$$

Шаг 5.

$$\text{GOTO}(I_6, L) = \text{CLOSURE}(\{[S \rightarrow F \wedge L \cdot]\}) = \{[S \rightarrow F \wedge L \cdot]\} = I_{10}, \Delta = \Delta \cup \{\delta(I_6, L) = \{I_{10}\}\}$$

$$\text{GOTO}(I_6, j) = \text{CLOSURE}(\{[L \rightarrow j \cdot]\}) = \{[L \rightarrow j \cdot]\} = I_9, \Delta = \Delta \cup \{\delta(I_6, j) = \{I_9\}\}$$

$$\phi_0 = \phi_0 \cup \{I_{10}\} \cup \{I_9\} = \{I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}\}$$

Шаг 6.

$$\text{GOTO}(I_7,) = \text{CLOSURE}(\{[S \rightarrow (S) \cdot]\}) = \{[S \rightarrow (S) \cdot]\} = I_{11}, \Delta = \Delta \cup \{\delta(I_7,) = \{I_{11}\}\}$$

$$\phi_0 = \phi_0 \cup \{I_{11}\} = \{I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11}\}$$

Результат:

$$\phi = \phi_0 = \{$$

$$I_0 = \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\},$$

$$I_1 = \{[S' \rightarrow S \cdot]\},$$

$$I_2 = \{[S \rightarrow F \cdot \wedge L]\},$$

$$I_3 = \{[S \rightarrow (\cdot S)], [S \rightarrow F \wedge \cdot L], [S \rightarrow (S) \cdot], [F \rightarrow \& \cdot L], [F \rightarrow i \cdot]\},$$

$$I_4 = \{[F \rightarrow \& \cdot L], [L \rightarrow \cdot j]\},$$

$$I_5 = \{[F \rightarrow i \cdot]\},$$

$$I_6 = \{[S \rightarrow F \wedge \cdot L], [L \rightarrow \cdot j]\},$$

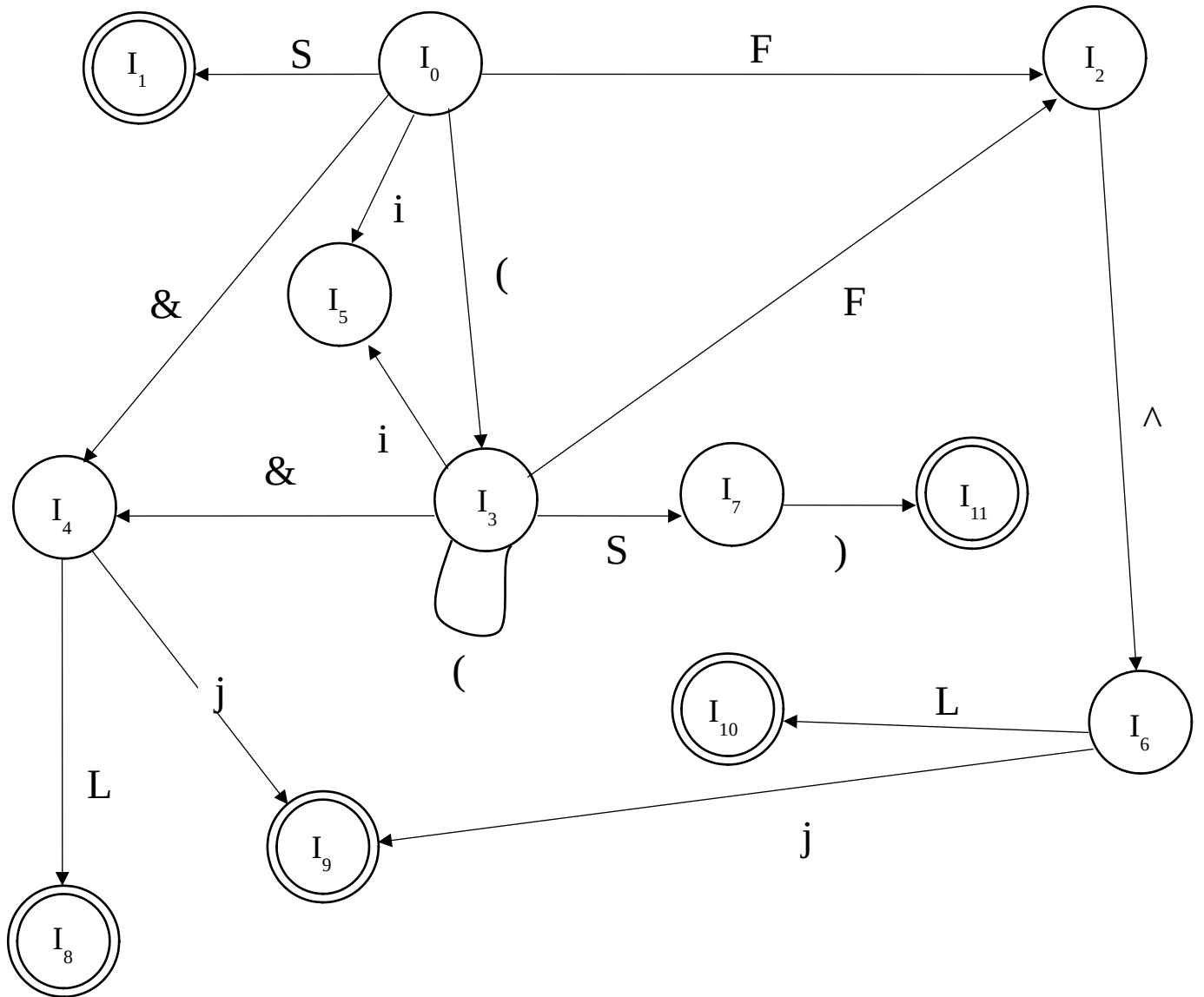
$$I_7 = \{[S \rightarrow (S) \cdot]\},$$

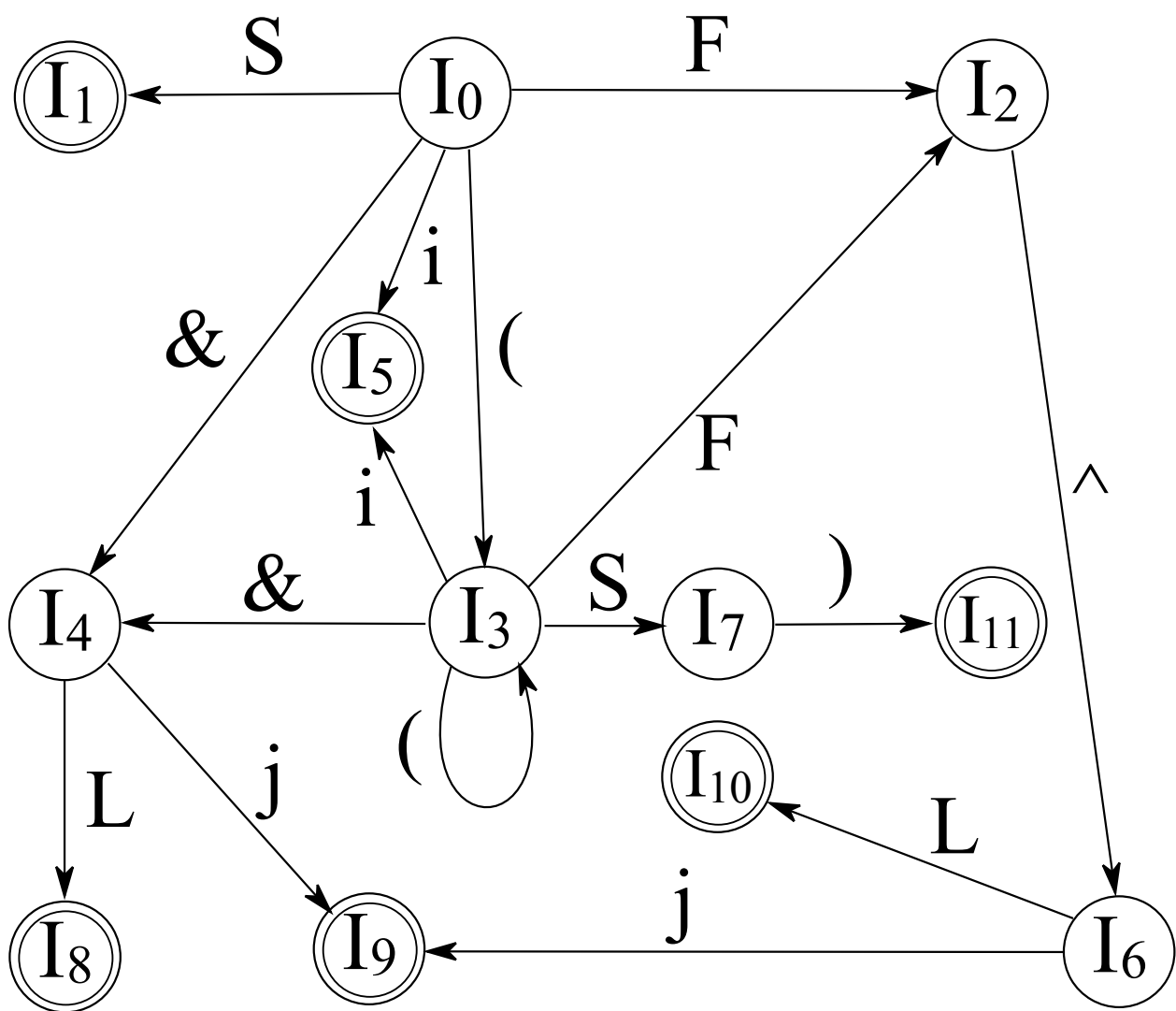
$$I_8 = \{[F \rightarrow \&L \cdot]\},$$

$$I_9 = \{[L \rightarrow j \cdot]\},$$

$$I_{10} = \{[S \rightarrow F \wedge L \cdot]\},$$

$$I_{11} = \{[S \rightarrow (S) \cdot]\}$$

$$q_f = \{I_1, I_8, I_9, I_{10}, I_{11}\}$$




Алгоритм: построение управляющей таблицы КС-грамматики G'

Вход: пополненная КС-грамматика $G'=(T, V', P', S')$, канонический набор множеств LR(0)-ситуаций ϕ , LR(0)-автомат переходов $KA=(Q=\phi, \Sigma=T \cup V' \cup \{\$, \delta, I_0, q_f)$

Выход: управляющая таблица M для пополненной КС-грамматики G'

Таблица M задаёт отображение множества $\phi \times \Sigma$ в множество, состоящее из следующих элементов:

3. СВЁРТКА ;
4. ДОПУСК ;
5. ПЕРЕНОС ;
6. j – номер состояния для перехода.

```
foreach (  $I_i \in \phi, i \leq |\phi|$  ) {
    foreach(  $\{ \delta(I_i, a) = \{ I_j \} \} \in \Delta$  ) {
        if (  $I_j \in q_f$  ) {
            foreach(  $[ A \rightarrow \alpha \cdot ] \in I_j$  ) {
                if (  $A \neq S'$  ) {
                    // Для просмотра следующего символа при
                    // SLR(1) берутся все символы из FOLLOW(A)
                    // То есть будет foreach(  $a \in \text{FOLLOW}(A)$  )
                    foreach (  $a \in T$  ) {
                         $M[j, a] = \text{СВЁРТКА } A \rightarrow \alpha$ 
                    }
                } else {
                     $M[j, \$] = \text{ДОПУСК}$ 
                }
            }
        }
        if(  $a \in T$  ) {
             $M[i, a] = \text{ПЕРЕНОС } j$ 
        }
        if(  $a \in V'$  ) {
             $M[i, a] = j$ 
        }
    }
}
```


Пример:

Пополненная грамматика $G' = (\{i, j, \&, ^, (,)\}, \{S', S, F, L\}, P', S')$

P' :

0. $S' \rightarrow S$
1. $S \rightarrow F \wedge L$
2. $S \rightarrow (S)$
3. $F \rightarrow \&L$
4. $F \rightarrow i$
5. $L \rightarrow j$

$$\phi = \phi_0 = \{$$

$$\begin{aligned} I_0 &= \{[S' \rightarrow \cdot S], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\}, \\ I_1 &= \{[S' \rightarrow S \cdot]\}, \\ I_2 &= \{[S \rightarrow F \cdot \wedge L]\}, \\ I_3 &= \{[S \rightarrow (\cdot S)], [S \rightarrow \cdot F \wedge L], [S \rightarrow \cdot (S)], [F \rightarrow \cdot \&L], [F \rightarrow \cdot i]\}, \\ I_4 &= \{[F \rightarrow \& \cdot L], [L \rightarrow \cdot j]\}, \\ I_5 &= \{[F \rightarrow i \cdot]\}, \\ I_6 &= \{[S \rightarrow F \wedge \cdot L], [L \rightarrow \cdot j]\}, \\ I_7 &= \{[S \rightarrow (S \cdot)]\}, \\ I_8 &= \{[F \rightarrow \&L \cdot]\}, \\ I_9 &= \{[L \rightarrow j \cdot]\}, \\ I_{10} &= \{[S \rightarrow F \wedge L \cdot]\}, \\ I_{11} &= \{[S \rightarrow (S) \cdot]\} \end{aligned}$$

Шаг 1. Сначала рассмотрим I_0 .

Ситуация $[S' \rightarrow \cdot S]$ даёт $M[0, S] = 1$, так как $GOTO(I_0, S) = I_1$.

Ситуация $[S \rightarrow \cdot F \wedge L]$ даёт $M[0, F] = 2$, так как $GOTO(I_0, F) = I_2$.

Ситуация $[S \rightarrow \cdot (S)]$ даёт $M[0, (] = \text{ПЕРЕНОС } 3$, так как $GOTO(I_0, () = I_3$.

Ситуация $[F \rightarrow \cdot \&L]$ даёт $M[0, \&] = \text{ПЕРЕНОС } 4$, так как $GOTO(I_0, \&) = I_4$.

Ситуация $[F \rightarrow \cdot i]$ даёт $M[0, i] = \text{ПЕРЕНОС } 5$, так как $GOTO(I_0, i) = I_5$.

Шаг 2. Рассмотрим I_1 : ситуация $[S' \rightarrow S \cdot]$ даёт $M[1, \$] = \text{ПРИНЯТИЕ}$.

Шаг 3. Рассмотрим I_2 .

Ситуация $[S \rightarrow F \cdot \wedge L]$ даёт $M[2, \wedge] = \text{ПЕРЕНОС } 6$, так как $GOTO(I_2, \wedge) = I_6$.

Шаг 4. Рассмотрим I_3 .

Ситуация $[S' \rightarrow (\cdot S)]$ даёт $M[3, S] = 7$, так как $GOTO(I_3, S) = I_7$.

Ситуация $[S \rightarrow \cdot F \wedge L]$ даёт $M[3, F] = 2$, так как $GOTO(I_3, F) = I_2$.

Ситуация $[S \rightarrow \cdot (S)]$ даёт $M[3, (] = \text{ПЕРЕНОС } 3$, так как $GOTO(I_3, () = I_3$.

Ситуация $[F \rightarrow \cdot \&L]$ даёт $M[3, \&] = \text{ПЕРЕНОС } 4$, так как $GOTO(I_3, \&) = I_4$.

Ситуация $[F \rightarrow \cdot i]$ даёт $M[3, i] = \text{ПЕРЕНОС } 5$, так как $GOTO(I_3, i) = I_5$.

Шаг 5. Рассмотрим I_4 .

Ситуация $[F \rightarrow \& \cdot L]$ даёт $M[4, L] = 8$, так как $GOTO(I_4, L) = I_8$.

Ситуация $[L \rightarrow \cdot j]$ даёт $M[4, j] = \text{ПЕРЕНОС } 9$, так как $GOTO(I_4, j) = I_9$.

Шаг 6. Рассмотрим I_5 .

Ситуация $[F \rightarrow i \cdot]$ даёт $M[5, \wedge] = \text{СВЁРТКА 4}$, так как $\text{FOLLOW}(F) = \{\wedge\}$ и продукция $F \rightarrow i$ имеет номер 4 в P' .

Шаг 7. Рассмотрим I_6 .

Ситуация $[S \rightarrow F \wedge L \cdot]$ даёт $M[6, L] = 10$, так как $\text{GOTO}(I_6, L) = I_{10}$.

Ситуация $[L \rightarrow \cdot j]$ даёт $M[6, j] = \text{ПЕРЕНОС 9}$, так как $\text{GOTO}(I_6, j) = I_9$.

Шаг 8. Рассмотрим I_7 .

Ситуация $[S \rightarrow (S \cdot)]$ даёт $M[7,)] = \text{ПЕРЕНОС 11}$, так как $\text{GOTO}(I_7,) = I_{11}$.

Шаг 9. Рассмотрим I_8 .

Ситуация $[F \rightarrow \& L \cdot]$ даёт $M[8, \wedge] = \text{СВЁРТКА 3}$, так как $\text{FOLLOW}(F) = \{\wedge\}$ и продукция $F \rightarrow \& L$ имеет номер 3 в P' .

Шаг 10. Рассмотрим I_9 .

Ситуация $[L \rightarrow j \cdot]$ даёт $M[9, \wedge] = \text{СВЁРТКА 5}$, $M[9,)] = \text{СВЁРТКА 5}$ и $M[9, \$] = \text{СВЁРТКА 5}$, так как $\text{FOLLOW}(L) = \{\wedge,), \$\}$ и продукция $L \rightarrow j$ имеет номер 5 в P' .

Шаг 11. Рассмотрим I_{10} .

Ситуация $[S \rightarrow F \wedge L \cdot]$ даёт $M[10,)] = \text{СВЁРТКА 1}$ и $M[10, \$] = \text{СВЁРТКА 1}$, так как $\text{FOLLOW}(S) = \{), \$\}$ и продукция $S \rightarrow F \wedge L$ имеет номер 1 в P' .

Шаг 12. Рассмотрим I_{11} .

Ситуация $[S \rightarrow (S) \cdot]$ даёт $M[11,)] = \text{СВЁРТКА 2}$ и $M[11, \$] = \text{СВЁРТКА 2}$, так как $\text{FOLLOW}(S) = \{), \$\}$ и продукция $S \rightarrow (S)$ имеет номер 2 в P' .

После выполнения всех вышеописанных шагов, получим управляющую таблицу M , приведённую ниже.

I	ACTION							GOTO		
	i	j	&	\wedge	()	\$	S	F	L
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			

Алгоритм: LR -анализ

Вход: входная строка w и таблица M LR-анализа с функциями ACTION и GOTO для пополненной КС-грамматики G' .

Выход: если $w \in L(G')$, шаги свёрток восходящего синтаксического анализа w ; в противном случае – указание о произошедшей ошибке.

Изначально в стеке синтаксического анализатора находится начальное состояние 0 , а во входном буфере $w\$$. Булевы переменные `accepted` и `error` отвечают за допуск входной цепочки языка, порождаемому грамматикой G' .

```
// a – первый символ цепочки w$
accepted = false;
error = false;
do {
    // s – состояние на вершине стека
    switch ( M[s, a] ) {
        case ДОПУСК :
            // Анализ завершён, цепочка допущена
            accepted = true
            break;
        case ПЕРЕНОС t :
            Внести t в стек
            Присвоить a следующий входной символ
            break;
        case СВЁРТКА  $A \rightarrow \beta$  :
            Снять  $|\beta|$  символов со стека
            // Теперь на вершине стека находится состояние t
            Внести в стек M[t, A]
            Вывести продукцию  $A \rightarrow \beta$ 
            break;
        default:
            // Анализ завершён, цепочка отвергнута
            error = true
            break;
    }
} while (!accepted И !error)
```

Пример:

Вывод цепочек:

1. $S \Rightarrow^1 F \wedge L \Rightarrow^3 \&L \wedge L \Rightarrow^5 \&j \wedge L \Rightarrow^5 \&j \wedge j$
2. $S \Rightarrow^2 (S) \Rightarrow^1 (F \wedge L) \Rightarrow^3 (\&L \wedge L) \Rightarrow^5 (\&j \wedge L) \Rightarrow^5 (\&j \wedge j)$
3. $S \Rightarrow^2 (S) \Rightarrow^2 ((S)) \Rightarrow^1 ((F \wedge L)) \Rightarrow^4 ((i \wedge L)) \Rightarrow^5 ((i \wedge j))$

Такты работы алгоритма, распознавание цепочек:

1. $(0, \&j \wedge j \$, \varepsilon) \vdash^\Pi (0\ 4, j \wedge j \$, \varepsilon) \vdash^\Pi (0\ 4\ 9, \wedge j \$, \varepsilon) \vdash^C (0\ 4\ 8, \wedge j \$, 5) \vdash^C (0\ 2, \wedge j \$, 35) \vdash^\Pi$
 $\vdash^\Pi (0\ 2\ 6, j \$, 35) \vdash^\Pi (0\ 2\ 6\ 9, \$, 35) \vdash^C (0\ 2\ 6\ 10, \$, 535) \vdash^C (0\ 1, \$, 1535) \vdash^\Delta$
 $\vdash^\Delta \text{ДОПУСК}$
2. $(0, (\&j \wedge j) \$, \varepsilon) \vdash^\Pi (0\ 3, \&j \wedge j) \$, \varepsilon) \vdash^\Pi (0\ 3\ 4, j \wedge j) \$, \varepsilon) \vdash^\Pi (0\ 3\ 4\ 9, \wedge j) \$, \varepsilon) \vdash^C$
 $\vdash^C (0\ 3\ 4\ 8, \wedge j) \$, 5) \vdash^C (0\ 3\ 2, \wedge j) \$, 35) \vdash^\Pi (0\ 3\ 2\ 6, j) \$, 35) \vdash^\Pi (0\ 3\ 2\ 6\ 9,) \$, 35) \vdash^C$
 $\vdash^C (0\ 3\ 2\ 6\ 10,) \$, 535) \vdash^C (0\ 3\ 7,) \$, 1535) \vdash^\Pi (0\ 3\ 7\ 11, \$, 1535) \vdash^C$
 $\vdash^C (0\ 1, \$, 21535) \vdash^\Delta \text{ДОПУСК}$
3. $(0, ((i \wedge j)) \$, \varepsilon) \vdash^\Pi (0\ 3, (i \wedge j)) \$, \varepsilon) \vdash^\Pi (0\ 3\ 3, i \wedge j) \$, \varepsilon) \vdash^\Pi (0\ 3\ 3\ 5, \wedge j) \$, \varepsilon) \vdash^C$
 $\vdash^C (0\ 3\ 3\ 2, \wedge j) \$, 4) \vdash^\Pi (0\ 3\ 3\ 2\ 6, j) \$, 4) \vdash^\Pi (0\ 3\ 3\ 2\ 6\ 9,)) \$, 54) \vdash^C$
 $\vdash^C (0\ 3\ 3\ 2\ 6\ 10,)) \$, 54) \vdash^C (0\ 3\ 3\ 7,)) \$, 154) \vdash^\Pi (0\ 3\ 3\ 7\ 11,) \$, 154) \vdash^C$
 $\vdash^C (0\ 3\ 7,) \$, 2154) \vdash^\Pi (0\ 3\ 7\ 11, \$, 2154) \vdash^C (0\ 1, \$, 22154) \vdash^\Delta \text{ДОПУСК}$

Выводы

В процессе выполнения курсовой работы я изучил алгоритм построения LR - анализатора на основе множеств CLOSURE и GOTO , составил программу на языке C#, моделирующую LR(0) -анализатор.

Во время изучения материала я столкнулся с проблемой перевода, пришлось обратиться к английским источникам, чтобы лучше понять материал. Дополнительно я изучил алгоритмы LL -анализа и LR -анализа.

Основной сложностью было внедрить свой код в уже имеющиеся наработки, с чем помог справиться GitHub и документация Doxygen. Система контроля версий помогла отслеживать изменения в большом коде, что сэкономило время.

Я познакомился с новым для себя языком программирования – C#. Он оказался очень похож на Java, с которым я немного работал, поэтому получилось быстро разобраться.

LR -анализатор является наиболее распространённым типом восходящего синтаксического анализа на сегодняшний день, полученные знания помогут мне, если в дальнейшей работе я столкнусь с ним.

Список литературы

1. Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман. *Компиляторы: принципы, технологии и инструментарий, 2-е издание.* – Издательский дом «Вильямс», 2008. Перевод с английского: И. В. Красиков. – 1184 с. (ISBN 978-5-8459-1349-4 (рус.))
2. *LL(k)-грамматики, множества FIRST и FOLLOW*
URL: [https://neerc.ifmo.ru/wiki/index.php?title=LL\(k\)-грамматики, множества FIRST и FOLLOW](https://neerc.ifmo.ru/wiki/index.php?title=LL(k)-грамматики, множества FIRST и FOLLOW) (дата обращения 21.05.2021)
3. *LR(k)-грамматики*
URL: [https://neerc.ifmo.ru/wiki/index.php?title=LR\(k\)-грамматики](https://neerc.ifmo.ru/wiki/index.php?title=LR(k)-грамматики) (дата обращения 21.05.2021)