

*Розглянутий метод низхідного синтаксичного аналізу з прогнозованим вибором альтернатив для контекстно-залежних граматики. Результати досліджень можуть застосовуватися при розробці граматичних і автоматних моделей мовних процесорів; при створенні сучасних методів проектування програмного забезпечення різного призначення*

*Ключові слова: синтаксичний аналіз, контекстно-вільні мови, контекстно-залежних мови мови, граматична модель*

*Rассмотрен метод нисходящего синтаксического анализа с прогнозируемым выбором альтернатив для контекстно-зависимых грамматик. Результаты исследований могут применяться при разработке грамматических и автоматных моделей языковых процессоров; при создании современных методов проектирования программного обеспечения различного назначения*

*Ключевые слова: синтаксический анализ, контекстно-свободные языки, контекстно-зависимых языки языки, грамматическая модель*

*The method of top-down syntactic analysis is considered with the forecast choice of alternatives for contextsensitive grammars. The results of researches can used at development of grammatical and automat models of language processors; at creation of modern methods of planning of different setting software*

*Key words: syntactic analysis, context-free languages, contextsensitive languages of language, grammatical*

# МЕТОД НИСХОДЯЩЕГО АНАЛИЗА С ПРОГНОЗИРУЕМЫМ ВЫБОРОМ АЛЬТЕРНАТИВ ДЛЯ КОНТЕКСТНО- ЗАВИСИМЫХ ГРАММАТИК

**В. В. Евсеев**

Кандидат технических наук, доцент\*

**З. В. Плотникова**

Кандидат технических наук, доцент\*

\*Кафедра ТАПР

Харьковский национальный университет

радиоэлектроники

пр. Ленина, 14, г. Харьков, 61166

Контактный тел.: 8 (057) 702-14-86

## 1. Введение

Одним из разделов технологии программирования является теория формальных языков, в которой особое внимание уделяется развитию контекстно-свободных (КС) и контекстно-зависимых (КЗ) языков, алгебрам языков, проблемам полноты этих алгебр и т.д. Результаты исследований в этой области применяются при разработке грамматических и автоматных моделей языковых процессоров, а также при создании современных методов проектирования программного и лингвистического обеспечения систем автоматизации проектных работ (САПР).

Исследованию формальных языков и грамматик, а также формализации синтаксиса языка посвящены работы [1],[3],[5],[7], в которых есть описание хорошо известных методов синтаксического анализа разверткой или построением вывода цепочки для различного класса грамматик.

Обычно, анализатор, основанный лишь на свертке или развертке, может использовать правила вывода, которые могут привести к тупиковым ситуациям. Для их преодоления, как правило, предусматривается возврат к некоторой метке, откуда можно выбрать новый вариант анализа. Такой возврат сопровождается восстановлением отдельных цепочек анализируемой цепочки (при свертке) и стиранием некоторых групп символов (при развертке).

## 2. Постановка задачи

С точки зрения практических приложений значительный интерес представляют методы беступикового анализа, при котором появление хотя бы одного тупика свидетельствует о том, что  $x \notin L(G)$ , где  $L(G)$  - язык, порожденный заданной грамматикой  $G$ , а  $x$  - цепочка [6].

В данной работе предлагается метод выбора стратегии при выводе цепочки  $x$ , основанный на построении таблиц истинности и не приводящий к тупиковым ситуациям при условии, что выводимая цепочка  $x \in L(G)$ , причем  $G$  в рассматриваемом случае является КЗ-грамматикой, или грамматикой общего вида.

Известно, что грамматика  $G$  называется контекстно-зависимой (или неукорачивающейся), если каждое правило имеет вид  $\alpha \rightarrow \beta$ , где  $\alpha, \beta$  обозначают цепочки, которые могут содержать как терминалы, так и нетерминалы, причем должно выполняться условие  $|\alpha| \leq |\beta|$ , удовлетворяющее требованию непустоты цепочки [1].

Пусть дана контекстно-зависимая грамматика  $G = (V_N, V_T, P, S)$ , где  $V_N$  и  $V_T$  - множества нетерминальных и терминальных символов,  $P$  - множество синтаксических правил грамматики,  $S$  - начальный нетерминальный символ.  $V_N = \{S, B, C\}$ ,  $V_T = \{a, b, c\}$ , множество правил  $P$ :

$$\left. \begin{aligned} p_1 : S &\rightarrow aSBC \vee abC \\ p_2 : CB &\rightarrow BC \\ p_3 : bB &\rightarrow bb \\ p_4 : bC &\rightarrow bc \\ p_5 : cC &\rightarrow cc \end{aligned} \right\} \quad (1)$$

Покажем, что используя предложенный авторами метод нисходящего анализа с прогнозируемым выбором альтернатив, можно заранее предсказать, какую альтернативу нужно будет выбирать на очередном шаге вывода в соответствии с первым символом из неп прочитанной части входной цепочки. Далее мы подробно рассмотрим этот подход.

**3. Метод нисходящего спуска с прогнозируемым выбором альтернатив**

Представим множество правил  $P$  в виде системы уравнений, одновременно произведя свертку цикла в первом уравнении по правилам вывода в алгебре рекурсивных событий [7]:

$$\left. \begin{aligned} S &= \{aabCBBC\} \\ CB &= BC \\ bB &= bb \\ bC &= bc \\ cC &= cc \end{aligned} \right\} \quad (2)$$

Связность символов левых частей уравнений системы уравнения (2) между собой приведена в таблице 1.

**Таблица 1**

Связность символов левых частей уравнений (2)

	C	B	a	b	c
C		+			
B					
a					
b	+	+			
c	+				

Построенная таблица 1 является базовой, не изменяющейся. Теперь, для определения порядка исключения неизвестных в полученной системе уравнений, необходимо построить таблицу 2 последовательных связностей между символами, входящими в правую часть первого уравнения в системе уравнений (2).

**Таблица 2**

Связность между символами, входящими в правую часть первого уравнения в системе уравнений (2)

	C	B	a	b	c
C		+			
B	+				
a			+	+	
b	+				
c					

Далее, сравнив таблицу 1 и таблицу 2 (базовую и полученную), увидим совпадение в двух ячейках: CB и bC. Так как нас интересует в первую очередь исключение неизвестных вида  $Q_i Q_j$ , где  $Q_i$  - нетерминалы, то остановим свой выбор на применении подстановки  $CB=BC$ .

Получаем

$$S = \{aabCBBC\} \xrightarrow{CB=BC} \{aabBCC\} \quad (3)$$

Аналогично таблице 2 строим следующую таблицу 3 попарных связностей между символами, входящими в правую часть полученного уравнения (3).

**Таблица 3**

Связность между символами, входящими в правую часть уравнения (3)

	C	B	a	b	c
C	+				
B	+				
a			+	+	
b		+			
c					

Сравниваем таблицу 3 с базовой таблицей 1 и определяем, что на следующем шаге необходимо применить подстановку  $bB=bb$ :

$$S = \{aabCBBC\} \xrightarrow{CB=BC} \{aabBCC\} \xrightarrow{bB=bb} \{aabbbCC\} \quad (4)$$

Далее, получаем таблицу 4:

**Таблица 4**

Связность между символами, входящими в правую часть выражения (4)

	C	B	a	b	c
C	+				
B					
a			+	+	
b	+			+	
c					

Сравнение таблицы 4 с основной таблицей показало, что совпадение в ячейке bC, следовательно применением подстановку bC=bc:

$$S = \{aab\underline{C}bC\} \xrightarrow{CB=BC} \{aabBCC\} \xrightarrow{bB=bb} \{aabbCC\} \xrightarrow{bC=bc} \{aabbC\} \quad (5)$$

Четвертая подстановка, найденная таким же путем построения таблицы связности и сравнение ее с основной таблицей 1, приводит к окончательному результату:

$$S = \{aab\underline{C}bC\} \xrightarrow{CB=BC} \{aabBCC\} \xrightarrow{bB=bb} \{aabbCC\} \xrightarrow{bC=bc} \{aabbC\} \xrightarrow{cC=cc} \{aabbcc\}. \quad (6)$$

Грамматика, на примере которой был рассмотрен предложенный авторами метод беступикового вывода цепочки, относится к типу КЗ-грамматик, которые содержат одинаковые по длине цепочки левых и правых частей вывода.

Задача беступикового вывода цепочки усложняется, если рассмотреть КЗ-грамматику с множеством других правил вывода (7):

$$\left. \begin{array}{l} p_1 : S \rightarrow ASD \\ p_2 : S \rightarrow ABa \\ p_3 : aD \rightarrow Ha \\ p_4 : H \rightarrow D \\ p_5 : B \rightarrow b \\ p_6 : bD \rightarrow bBa \\ p_7 : A \rightarrow a \end{array} \right\} \quad (7)$$

Аналогично вышерассмотренному случаю представим множество правил P в виде системы уравнений, одновременно произведя свертку цикла в первом уравнении. Находим систему уравнений, согласно правилам алгебры рекурсивных событий:

$$\left. \begin{array}{l} S = \{A\underline{A}BaD\} \\ aD = Ha \\ H = D \\ B = b \\ bD = bBa \\ A = a \end{array} \right\} \quad (8)$$

Используя вышеприведенный метод, необходимо учесть, что сочетания вида  $\emptyset S_i = S_i \emptyset = S_i$ , (где  $\emptyset$  - пустой нетерминал), используются для выравнивания длины левых частей уравнения. В результате выбранной стратегии подстановок по предложенному методу получаем результат:

$$S = \{A\underline{A}BaD\} \xrightarrow{A=a} \{a\underline{A}BaD\} \xrightarrow{B=b} \{a\underline{A}baD\} \xrightarrow{A=a} \{a\underline{a}baD\} \xrightarrow{aD=Ha} \{a\underline{a}baH\} \xrightarrow{H=D} \{a\underline{a}baD\} \xrightarrow{bD=bBa} \{a\underline{a}bba\} \xrightarrow{bD=bBa} \{a\underline{a}bba\} \quad (9)$$

#### 4. Выводы

Разработанный метод нисходящего анализа с прогнозируемым выбором альтернатив представляет интерес для класса КЗ-грамматик, так как считается, что не существует метода для получения безвозвратного вывода цепочек для контекстно-зависимых грамматик. Таблицы истинности удобны для формализации метода выбора последовательности подстановок при решении системы уравнений, представляющих КЗ-грамматики общего вида. Рассмотренный в статье метод выбора стратегии нельзя назвать методом выбора оптимальной стратегии, поскольку доказательства оптимальности не приведены, но приводящий, как правило, к беступиковому выводу цепочки  $x \in L(G)$  и являющийся на данный момент времени наиболее простым из известных методов выбора стратегии беступикового вывода.

#### Литература

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции.-М. – Мир, 1987.- Т.1,2.- 612с; 487с.
2. Бурбаки Н. Теория множеств. – М. – Мир, 1985.- 450с.
3. Вельбицкий И.В., Технология программирования.- Киев: Техника, 1994.-279с.
4. Лавров С.С., Программирование. Математические основы, средства, теория. СПб.: БХВ-Петербург, 2001
5. Ларин А.А., Березюк Н.Т., Водопянов В.К., Структурное проектирование автоматов и алгоритмов. М., Министерство обороны. 1994.-405с.
6. Шальто А.А., Туккель Н.И., Преобразование итеративных алгоритмов в автоматные//Программирование. 2002. №5.
7. Водопянов В.К., Плотникова З.В., Синтез аппаратных компиляторов. Деп. В ЦНТИ «Поиск», №035-4786, М., 1991.-120с.