



УДК 004.02

Составители: А.И. Катыхин, Е.Н. Иванова

Рецензент

Профессор кафедры биомедицинской инженерии,  
доктор технических наук

*С.А. Филист*

**Построение синтаксических анализаторов входных языков в САПР:** методические указания по выполнению лабораторной работы / Юго-Зап. гос. ун-т; сост.: А.И. Катыхин, Е.Н. Иванова. — Курск, 2017. — 13 с.: ил. 4. — Библиограф.: с. 13.

Рассматриваются задачи на составление грамматики для команд, операторов языков программирования. Приводится описание правил построения синтаксического графа на основе скобочной формы записи, на примерах объясняются основные приёмы записи грамматики в форме Бэкуса-Наура и предлагаются задачи для решения на занятии.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по направлению Информатика и вычислительная техника.

Предназначены для студентов направления 09.03.01 Информатика и вычислительная техника дневной формы обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.

Усл.печ.л. . Уч.-изд.л. . Тираж 20 экз. Заказ . Бесплатно.

Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## **1. Цель**

Ознакомление с принципами построения языковых анализаторов САПР, получение практических навыков в разработке средств синтаксического анализа и исследование возможности реализации грамматического разбора регулярных грамматик с помощью синтаксических таблиц.

## **2. Языки взаимодействия проектировщика с САПР**

Одной из важнейших задач при создании САПР является выбор языков взаимодействия и форм общения проектировщика с ЭВМ. Именно необходимостью в специальных языках взаимодействия человека с машиной САПР отличаются от чисто вычислительных систем.

Языки взаимодействия являются особыми, специально ориентированными на нужды проектировщика. Если языки программирования направлены, главным образом, на универсальность и удобство трансляции их в машинные коды, то языки взаимодействия предназначены для обеспечения наибольших удобств общения проектировщика с ЭВМ, наиболее компактного представления проектной информации и т.п.

Многообразие языков, используемых в САПР, можно разделить на следующие основные типы: естественный, ограниченный естественный, командный языки, «меню» и «шаблоны».

Естественный язык. Идея использования во взаимодействии с ЭВМ естественного языка (родного языка пользователя) в большинстве случаев не оправдывает себя не только из-за трудностей автоматического анализа сообщений пользователя, но и из-за того, что в пределах жестко ограниченной ситуации, связанной с задачами проектиро-

вания, использование всего богатства средств естественно-го для пользователя языка является нецелесообразным. Сам естественный язык при этом имеет тенденцию к сокращению словаря, упрощению синтаксиса вплоть до формализации. С учетом вышеуказанных замечаний естественный язык постепенно перерождается в ограниченно естественный язык.

Ограниченный естественный язык является перспективным способом взаимодействия. Основной недостаток подобного взаимодействия состоит в том, что пользователь должен хорошо представлять синтаксические семантические ограничения, накладываемые на естественный язык.

Командный язык является распространенным средством общения проектировщика с ЭВМ. Пользователю предоставляется фиксированный набор команд, с помощью которых происходит управление выполнением различных проектных процедур. Как правило, команды несут в себе два типа информации: определение процессов, подлежащих выполнению, и определение данных, подвергающихся обработке.

«Меню» и «шаблоны» являются наиболее распространенными средствами диалогового взаимодействия с прикладными программами САПР. С помощью «меню» пользователь управляет выполнением проектной процедуры, выбирая требуемую функцию из перечисленных в «меню». «Шаблон» представляет собой организованный специальным образом кадр диалога, предназначенный для ввода или вывода данных. Он содержит информационные поля, располагающиеся в определенных местах экрана, которые предназначены для обмена данными с пользователем. Однако взаимодействие только с использованием «меню» и «шаблонов» не обладает достаточной гибкостью и поэтому чаще всего применяется совместно с другими средствами общения ЭВМ и человека.

Используемые языки можно функционально разделить на три класса: входные, выходные и базовые.

Входные языки предназначены для описания проектируемых объектов и управления процедурами проектирования. Основным требованием, предъявляемым к входным языкам, является их максимальная близость к языку проектировщика в данной области.

Выходные языки ориентированы на вывод полученных в результате проектирования решений в виде необходимой проектной документации, удовлетворяющей требованиям изготовления объекта и стандартам.

Базовые языки способствуют осуществлению программирования основных процедур проектирования.

### **3. Основные положения теории формальных языков**

Рассмотрим основные понятия, которые используются в теории формальных языков.

Алфавит – непустое конечное множество элементов, называемых символами.

Цепочка – конечная последовательность символов алфавита. Например, если  $A = \{a, b, c\}$  – алфавит, то цепочками являются последовательности:  $a$ ,  $bc$ ,  $acsbac$ ,  $aa$  и т.п. Порядок следования символов в цепочке важен, поэтому цепочки  $ac$  и  $ca$  отличны друг от друга.

Длина цепочки – это число символов в цепочке, и обозначается как  $|X|$ . Так,  $|a| = 1$ ,  $|acsbac| = 6$ , а длина пустой цепочки равна нулю.

Конкатенация цепочек  $X$  и  $Y$  – это цепочка, полученная путем дописывания после символов цепочки  $X$  символов цепочки  $Y$ . Если  $X = ab$  и  $Y = cd$ , то  $XY = abcd$  и  $YX = cdab$ , а  $|XY| = |YX| = |X| + |Y|$ .

Правило подстановки – это упорядоченная пара  $Z = (U, X)$  записываемая как

$$U ::= X$$

и читаемая как «Символ  $U$  может состоять из цепочки  $X$ ».

Грамматика  $G(Z)$  – это конечное, непустое множество правил подстановки  $Z$ , где  $G$  – начальный символ, который должен встретиться в левой части, по крайней мере, одного правила подстановки.

Нетерминальные и терминальные символы, используемые в левых и правых частях правил подстановки образуют словарь. Нетерминальные встречаются как в левых, так и в правых частях правил, а терминальные символы только в правых. Таким образом, словарь  $U$  есть

$$U ::= U_T \cup U_N,$$

где  $U_T$  – множество терминальных символов,

$U_N$  – множество нетерминальных символов

Причем  $U_T \cap U_N = \emptyset$ , а при записи нетерминальные символы заключаются в угловые скобки, например,  $\langle \text{параметр} \rangle$ .

Множество правил  $U ::= X$ ,  $U ::= Y$ ,  $U ::= Z$  с одинаковыми левыми частями сокращенно можно записать так

$$U ::= X|Y|Z.$$

Форма записи с использованием двух последних замечаний называется нормальной формой Бэкуса или сокращенно БНФ. Таким образом, БНФ для грамматики целого числа без знака может выглядеть следующим образом:

$$\langle \text{число} \rangle ::= \langle \text{чис} \rangle \quad (1)$$

$$\langle \text{чис} \rangle ::= \langle \text{чис} \rangle | \langle \text{чис} \rangle \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \quad (2)$$

$$\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \quad (3)$$

Основное назначение грамматики – порождение синтаксически правильных предложений языка.

Цепочка  $U$  непосредственно порождает цепочку  $W$  (иначе: « $W$  непосредственно выводится из  $U$ »)  $U \Rightarrow W$ , если для цепочек  $X$  и  $Y$  и правила подстановки  $U ::= A$  можно записать:

$$U = XYU \text{ и } W = XYA.$$

Для порождения  $U \Rightarrow U_1 \Rightarrow U_2 \Rightarrow \dots \Rightarrow W$  можно кратко записать:  $U \xRightarrow{+} W$ , а если одновременно  $U \Rightarrow W$  и  $U \xRightarrow{+} W$ , то это обозначается как  $U \xRightarrow{*} W$ .

Цепочка  $X$  является сентенциальной формой, если  $X$  выводима из начального символа грамматики  $G(Z)$ , т.е.  $Z \xRightarrow{*} X$ .

Языком  $L$  называется множество предложений, порождаемых грамматикой  $G(Z)$ .

Грамматика рекурсивна по отношению к  $U$ , если  $U \xRightarrow{+} \dots U \dots$ . Возможны как левая, так и правая рекурсии. Правила подстановки также могут быть лево- и праворекурсивными. В качестве примера можно отметить леворекурсивное правило (2) из грамматики целого числа без знака.

#### 4. Способы представления грамматик языков

Существуют различные способы представления грамматик языков, которые отличаются наглядностью и областью применения.

Упомянутая выше нормальная форма Бэкуса имеет слабую наглядность, хотя и используется в технических документациях при описании различных языков программирования.

Аналогичную область применения имеет скобочная форма записи. При этом используются следующие обозначения:

- = – соответствие левой и правой частей,
- | – возможность выбора одного из символов,
- { } – указание альтернативных конструкций,
- [ ] – указание необязательных конструкций,
- ... – многократное повторение предыдущего элемента  
(количество повторений не указывается).

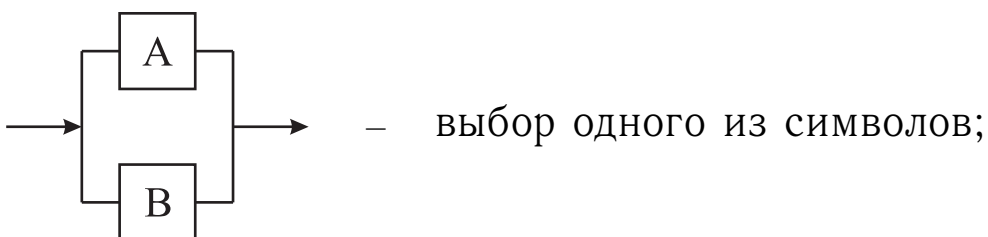
Кроме того, скобочная форма записи допускает использование некоторых правил и тождеств, а именно:

- а)  $\{AC|BC\} = \{A|B\}C$ ,
- б)  $\{AB|AC\} = A\{B|C\}$ ,
- в)  $\{A\dots\} = A[A]$ ,
- г)  $\{A|\emptyset\} = [A]$ .

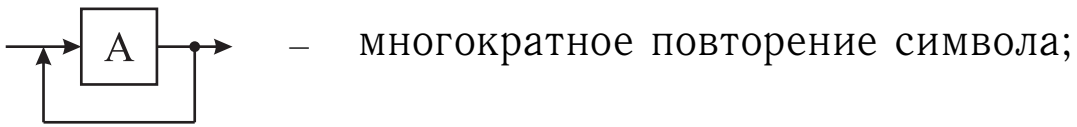
Так, например, грамматика целого числа в скобочной форме будет выглядеть следующим образом:

$$\langle \text{целое число} \rangle = [\{+|- \}]\{\{0|1|2|3|4|5|6|7|8|9\}\dots\}.$$

Наиболее наглядным способом представления грамматики по сравнению с предыдущими является синтаксический граф, при составлении которого используют следующие условные обозначения:







Синтаксический граф целого числа имеет вид, представленный на рис. 1.

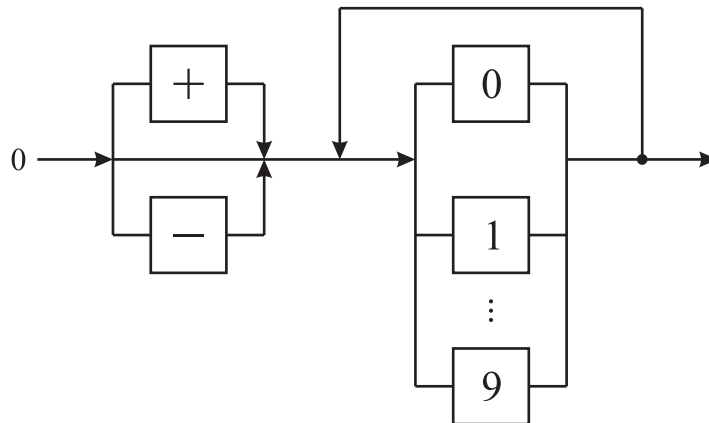


Рисунок 1 – Синтаксический граф целого числа

Для перехода от скобочной формы записи к синтаксическому графу используют правила, приведенные на рис. 2.

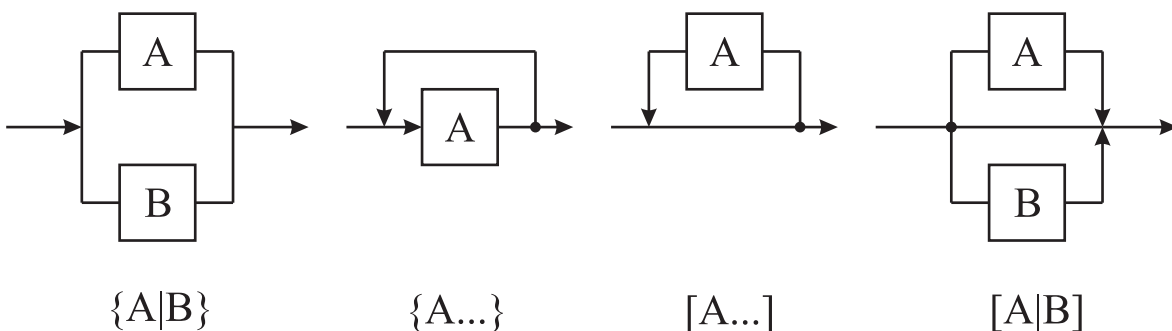


Рисунок 2 – Типовые фрагменты перехода от скобочной формы к синтаксическому графу

Преобразование синтаксического графа в БНФ происходит в два этапа. На первом синтаксический граф размеча-

- ется нетерминальными символами по следующим правилам:
- а) начальная вершина графа помечается начальным символом грамматики (рис. 3а);
  - б) между двумя терминальными символами обязательно присутствует хотя бы один нетерминальный символ (рис. 3б);
  - в) перед выбором нескольких терминальных символов ставится один нетерминальный (рис. 3в);
  - г) итерации (повторение) терминальных символов соответствует один нетерминальный (рис. 3г).

На втором этапе исходя из полученного размеченного графа записываются правила подстановки с использованием типовых конструкций перехода, изображенных на рис. 4.

Значительно упрощает процедуру синтаксического анализа, а так же удобна для представления в машинной форме синтаксическая таблица, построение которой сводится к по-

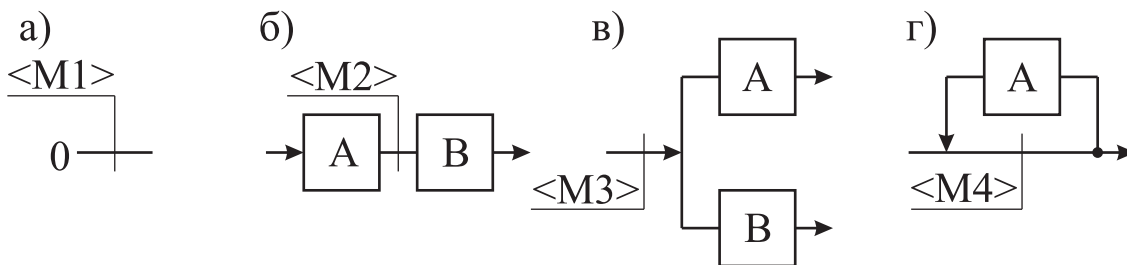


Рисунок 3 – Правила разметки синтаксического графа

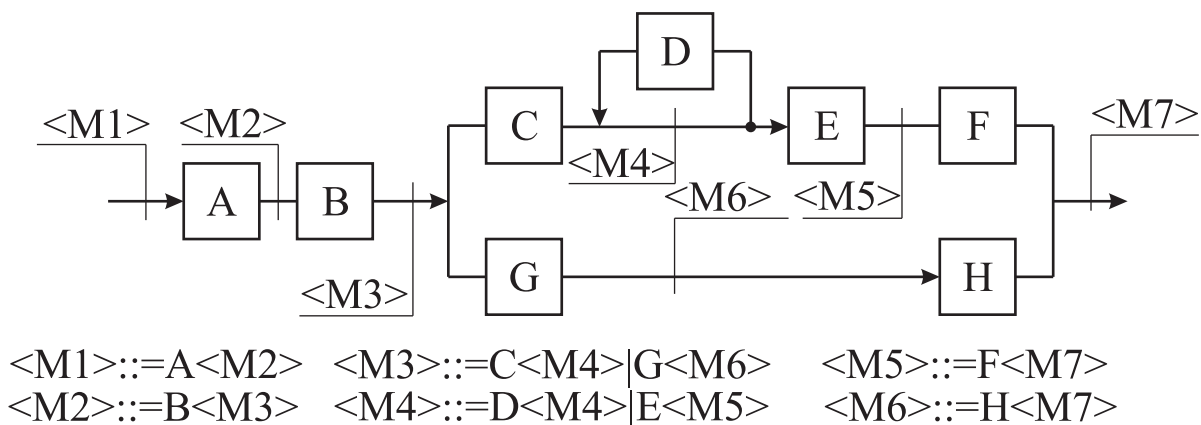


Рисунок 4 – Способы записи правил подстановки

следовательно обходу всех ветвей размеченного синтаксического графа.

По вертикали располагаются нетерминальные символы, определяющие текущее состояние синтаксического анализатора. Все состояния пронумерованы так, что одни из номеров присвоен состоянию «Неправильный синтаксис», а другой – «Нормальное завершение анализа». По горизонтали располагаются терминальные символы, которые определяют новое состояние анализатора. Кроме того, множество терминальных символов дополняется символом, имеющим условное название «Любой другой символ». Он соответствует тем символам, которые не употребляются при записи анализируемой грамматики. На пересечении вертикалей и горизонталей таблицы проставляются номера новых состояний синтаксического анализатора.

## **5. Порядок выполнения работы**

1. При подготовке к выполнению настоящей работы изучить материалы методических указаний к работе.
2. Провести анализ предложений грамматики, для чего составить конечное множество различных видов конструкций заданного оператора (команды). В ходе анализа следует руководствоваться следующими ограничениями:
  - а) при использовании арифметических выражений иметь в виду только операции сложения, вычитания, умножения и деления;
  - б) при использовании массивов в качестве его индексов принимать имена простых переменных (но не массивов) или целые числа;
  - в) следует учесть, что во многих языках программирования операторы могут иметь метки или нумерацию.

3. Получить скобочную форму записи грамматики оператора (команды).
4. Построить синтаксический граф на основе скобочной формы записи.
5. Составить синтаксическую таблицу, используя полученные ранее БНФ и синтаксический граф.
6. Разработать алгоритм, составить программу синтаксического анализатора на любом языке высокого уровня.
7. Оформить отчет по выполненной работе.

## **6. Задания для самостоятельного выполнения**

1. Объявление переменной целого типа.
2. Объявление переменной вещественного типа.
3. Операция инициализации переменной целого типа.
4. Операция инициализации переменной вещественного типа.
5. Объявление массива целых чисел.
6. Арифметическая («+», «-», «x», «/») операция с переменными.
7. Логическая («AND», «OR») операция с переменными.
8. Арифметическая («+», «-», «x», «/») операция переменной и числовой константы.
9. Логическая («AND», «OR») операция переменной и числовой константы.
10. Объявление функции.

## **7. Содержание отчета**

1. Формулировка задания.
2. Результаты анализа предложений языка.
3. Скобочная форма записи.

4. Синтаксический граф заданного оператора (команды)
5. Синтаксический граф, размеченный нетерминальными символами.
6. Грамматика заданного оператора (команды), записанная в БНФ.
7. Результаты выполнения программы.

## **8. Литература**

1. Серебряков, В.А. Основы конструирования компиляторов [Текст] / В.А. Серебряков, М.П. Галочкин. — М.: УРСС, 2001. — 174 с.
2. Фомичев, В.С. Формальные языки, грамматики и автоматы [Электронный ресурс] // Режим доступа — <http://old.eltech.ru/misc/LGA/2007/FINAL/Index.html>.