Министерство образования и науки Российской Федерации
ФГАО УВО “Севастопольский государственный университет”

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению самостоятельной работы по дисциплине
“Теоретические основы построения компиляторов”
для студентов всех форм обучения основного профиля 09.03.02 –
“Информационные системы и технологии”

Севастополь
2015

УДК 004.4

Методические указания к выполнению самостоятельной работы по дисциплине “Теоретические основы построения компиляторов” для студентов всех форм обучения основного профиля 09.03.02 – “Информационные системы и технологии” [Текст] / Разраб. В.Ю. Карлусов. – Севастополь: Изд-во СевГУ, 2015. – 56 с.

Цель методических указаний: Обеспечение студентов всех форм обучения дидактическим материалом для качественного выполнения лабораторных и контрольных работ; по разделам: "формальные языки и грамматики", "элементы теории конечных автоматов", "отношения предшествования" в приложении к задачам синтаксического анализа программ на алгоритмических языках.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных систем,
протокол № *01* от *03 февраля* 20*15* г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент Кожаев Е.А., кандидат техн. наук, доцент кафедры кибернетики и вычислительной техники.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В основу настоящей методической разработки положен личный опыт авторов, приобретённый в процессе параллельного с лекционным курсом и вычислительным практикумом на ЭВМ [1], проведения практических занятий по дисциплине. Сложившаяся ситуация диктовала необходимость дозированной подачи материала: во - первых, чтобы практическое занятие не превращалось в подобие лекции, во - вторых, требовалось закреплять отдельные теоретические положения, существенно важные для обеспечения плодотворной работы с ЭВМ.

В значительной мере, препятствием к проведению занятий на должном уровне являлось отсутствие сборника упражнений, направленных на закрепление теоретических положений дисциплины как аудиторно, так и самостоятельно. Это послужило стимулом к поиску как готовых задач и примеров, разбираемых в литературе, так и разработку своих собственных упражнений. Попутно проводилась адаптация хода решения отдельных задач для адекватного понимания его студентами (в первую очередь заочной формы обучения, действующими в отрыве от преподавателя). Указанный подход нашёл своё методическое оформление в [2], которое прошло апробацию именно на этой категории учащихся.

Настоящие методические указания призваны, в полном объёме, обеспечить учащихся фондом задач для самостоятельной работы. Для удобства восприятия и с целью методической подачи материала, упражнения сгруппированы по темам, в пределах каждой из групп поставлены цели. Приводимые упражнения снабжены ссылками на источник, если задание или условие было заимствовано. Последовательность подачи материала может варьироваться преподавателем, ведущим занятия. Рекомендуется следующий порядок изучения:

а) элементы теории формальных грамматик;

б) различные способы построения конечных автоматов;

в) минимизация конечных автоматов и построение детерминированных автоматов по недетерминированным конечным автоматам;

г) контрольная работа по теории конечных автоматов;

д) исследование грамматик на принадлежность к $LL(k)$ и построение нисходящих распознавателей;

е) исследование на отношение предшествования;

ж) эквивалентные преобразования грамматик.

1. ФОРМАЛЬНЫЕ ГРАММАТИКИ И ИХ ЭЛЕМЕНТЫ

Цели и задачи раздела — научиться:

а) определять терминальные, нетерминальные символы грамматики и языка, порождаемого ею, по системе продукции;

б) по предложениям языка построить соответствующую порождающую формальную грамматику;

в) проводить исследования грамматики на однозначность;

г) выполнять эквивалентные преобразования грамматики с целью придания необходимых свойств;

д) проводить классификацию формальных грамматик;

е) по круговой диаграмме алгоритмического языка строить формальную грамматику, его порождающую.

Система ссылок на теоретические источники

Теоретические положения, подлежащие усвоению	Исчерпывающее изложение	Доходчивое изложение	Изложение, адаптированное к восприятию
1. Элементы формальных грамматик	[3]	[5, 6, 10]	[2, с. 5]
2. Однозначность грамматики	[3, 6]	[6]	[2, с. 6 - 8]
3. Цепочки и предложения языка	[3 - 6]	[6]	[2, с. 5]
4. Классификация формальных грамматик	[3 - 6, 8 - 10]	[3 - 5, 8 - 12]	[2, с. 6]
5. Эквивалентные преобразования грамматик	[10]	[10]	[2, с. 37 - 41]

Примеры выполнения заданий

Определить элементы грамматики: словари терминалов и нетерминалов, аксиому и язык, порождаемый грамматикой:

$\mathbf{S} \rightarrow b\mathbf{A}|\mathbf{A}$;

$\mathbf{A} \rightarrow a|a\mathbf{A}$.

$V_N = \{\mathbf{S}, \mathbf{A}\}$ – словарь нетерминалов, $V_T = \{a, b\}$ – словарь терминалов, аксиомой грамматики является нетерминал **S**.

Для определения языка необходимо построить выводы цепочек терминальных символов:

$\mathbf{S} \Rightarrow b\mathbf{A} \Rightarrow ba$;

$\mathbf{S} \Rightarrow b\mathbf{A} \Rightarrow ba\mathbf{A} \Rightarrow \ldots \Rightarrow baa\ldots a\mathbf{A} \Rightarrow baa\ldots aa$;

$\mathbf{S} \Rightarrow \mathbf{A} \Rightarrow a$;

$\mathbf{S} \Rightarrow \mathbf{A} \Rightarrow a\mathbf{A} \Rightarrow \ldots \Rightarrow aa\ldots a\mathbf{A} \Rightarrow aa\ldots aa.$

Отсюда несложно видеть, что язык, порождаемый данной грамматикой, есть $L\{G[S]\} = \{a^k, a^k b,\ k \geq 1\}$.

Построить контекстно-свободную грамматику, порождающую следующие предложения языка $L = \{c(ab)^n d^k (ba)^n c,\ n, k \geq 1\}$.

Анализ предложений языка показывает, что сентенциальные формы обязательно начинаются и оканчиваются символом c. Следовательно, правило грамматики, определяющее аксиому, имеет вид $\mathbf{Z} \to c\mathbf{S}c$. Центральная часть цепочки неоднородна, её голова и хвост находятся в зависимости от n и конкатенируются одновременно, отсюда просматривается правило $\mathbf{S} \to ab\mathbf{S}ba$. При некотором значении n процесс останавливается, и выполняется генерация подцепочки d^k, что свидетельствует о наличии совокупности правил: $\mathbf{S} \to \mathbf{D}$ и $\mathbf{D} \to \mathbf{D}d|d$. Набор всех вышеизложенных правил является одним из возможных решений примера, то есть

$\mathbf{Z} \to c\mathbf{S}c$

$\mathbf{S} \to ab\mathbf{S}ba$

$\mathbf{S} \to \mathbf{D}$

$\mathbf{D} \to \mathbf{D}d|d$

Построить синтаксическое дерево для цепочки baa, ***порождённой грамматикой, представленной правилами***

$\mathbf{S} \to b\mathbf{A}|\mathbf{A}$

$\mathbf{A} \to a|a\mathbf{A}$

Процесс вывода удобно представлять в виде ориентированного графа, который называется *синтаксическим деревом* или *деревом вывода*. Для сентенциальной формы $\mathbf{S} \Rightarrow^* baa$, дерево вывода имеет вид

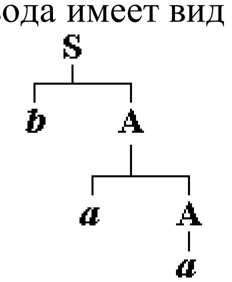

Рисунок 1.1 – Синтаксическое дерево

Является ли грамматика* $\mathbf{S}\rightarrow\mathbf{S}b\mathbf{S}|\mathbf{S}c\mathbf{S}|a$ *однозначной?

По определению, грамматика является *однозначной*, если *для каждой* сентенциальной формы существует *единственное* синтаксическое дерево, т.е., любая сентенциальная форма выводится единственным образом. Подчеркнём, что *порядок* применения правил *не существенен.*

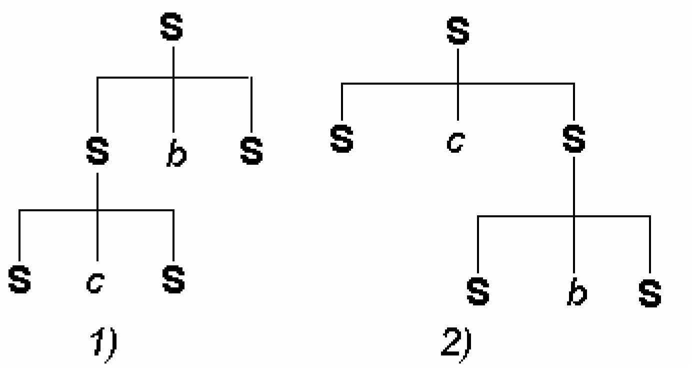

Рисунок 1.2 – Синтаксические деревья цепочки $\mathbf{S}c\mathbf{S}b\mathbf{S}$

Цепочке $\mathbf{S}c\mathbf{S}b\mathbf{S}$ соответствуют два синтаксических дерева. Следовательно, грамматика неоднозначна.

***Построить ε-свободную грамматику, эквивалентную заданной*.**

Пусть множество исходных правил имеет вид:

$P=\{\mathbf{S} \rightarrow \mathbf{[E]}|\mathbf{E};\ \mathbf{E} \rightarrow \mathbf{D}|\mathbf{D}\ \mathbf{+E}\ |\mathbf{D\text{-}E};\ \mathbf{D} \rightarrow \mathbf{F}|\mathbf{D}\ \ \mathbf{*F}|\mathbf{D}\ /\ \mathbf{F};\ \mathbf{F}\rightarrow a|b|c|\varepsilon\}$

1. Объединим все продукции, непосредственно порождающие ε, в множество P_ε.

$P_\varepsilon = \{\mathbf{F}\rightarrow\varepsilon\}$.

2. Отыщем все ε-порождающие нетерминальные символы $\mathbf{A}_i \in V_N$, такие что $\mathbf{A}_i \Rightarrow^* \varepsilon$ или $\mathbf{A}_i \Rightarrow^+ \varepsilon$. Для этого необходимо попытаться построить синтаксические выводы.

Имеем: $\mathbf{F} \Rightarrow \varepsilon$, $\mathbf{S} \Rightarrow^* \varepsilon$, $\mathbf{E} \Rightarrow^+ \varepsilon$, $\mathbf{D} \Rightarrow^+ \varepsilon$.

3. Каждой продукции из множества первичных правил, у которой в правой части находится ε-порождающий символ, поставим в соответствие продукцию, в правой части которой, по сравнению с исходной, опущен один или более нетерминальных символов. Эти продукции образуют множество P_0.

$P_0=\{\mathbf{S}\rightarrow[];\ \mathbf{E}\rightarrow\mathbf{D}+|+\ \mathbf{E}|\mathbf{D}\ –|–\ \mathbf{E}|+|–;\ \mathbf{D}\rightarrow*\ \mathbf{F}|\mathbf{D}\ *|\mathbf{D}\ /|/\ \mathbf{F}|/|*;\ \mathbf{F}\rightarrow a|b|c\}$.

4 Объединяя совокупности правил $P\backslash P\varepsilon \cup P_0$ получим продукции преобразованной грамматики:

$P=\{\mathbf{S} \rightarrow \mathbf{[E]}|\mathbf{E}|[];$

$\mathbf{E} \rightarrow \mathbf{D}|\mathbf{D} + \mathbf{E}|\mathbf{D} – \mathbf{E}|\ \mathbf{D} +|+\ \mathbf{E}|\mathbf{D}–|–\mathbf{E}|+|–;$

$\mathbf{D}\rightarrow\mathbf{F}|\mathbf{D}\ *\ \mathbf{F}|\mathbf{D}\ /\ \mathbf{F}|*\ \mathbf{F}|\ \mathbf{D}\ *|\mathbf{D}\ /|/\ \mathbf{F}|/|*;$

$\mathbf{F}\rightarrow a|b|c\}$.

Привести грамматику к нормальной форме Хомского

Для грамматики в нормальной форме Хомского, продукции имеют вид:

$$\mathbf{A} \to \mathbf{BC} \text{ или } \mathbf{A} \to a, \text{ где } \mathbf{A}, \mathbf{B}, \mathbf{C} \in V_N,\ a \in V_T.$$

Продемонстрируем процесс преобразования, используя алгоритм и пример из [10]. Пусть грамматика определяется совокупностью правил:

$$\begin{aligned}&\mathbf{S} \to \mathbf{ABA}|\mathbf{A}\\&\mathbf{A} \to a|a\mathbf{A}|\mathbf{B}\\&\mathbf{B} \to b|b\mathbf{B}.\end{aligned}$$

1. Отыщем все *первичные* продукции, у которых в правой части имеется единственный нетерминальный символ, вида $\mathbf{A}_i \to \mathbf{B}_j$, $\mathbf{A}_i$, $\mathbf{B}_j \in V_N$. Поставим в соответствие символу $\mathbf{A}_i$ совокупность непервичных правил, обеспечивающих вывод терминальной цепочки с использованием терминала $\mathbf{B}_j$ (через $\mathbf{B}_j$).

Первичные продукции, суть $\mathbf{S} \to \mathbf{A}$, $\mathbf{A} \to \mathbf{B}$. Правилу $\mathbf{S} \to \mathbf{A}$ ставим в соответствие правило $\mathbf{S} \to a\mathbf{A}|b\mathbf{B}|a|b$, а правилу $\mathbf{A} \to \mathbf{B}$ ставим в соответствие $\mathbf{A} \to b\mathbf{B}|b$. Получим

$$\begin{aligned}&\mathbf{S} \to \mathbf{ABA}|a\mathbf{A}|b\mathbf{B}|a|b\\&\mathbf{A} \to a\mathbf{A}|b\mathbf{B}|a|b\\&\mathbf{B} \to b|b\mathbf{B}.\end{aligned}$$

2. Отыщем *вторичные* продукции. Их правые части – цепочка длины более одного символа, содержащая подцепочки терминальных символов. Для каждого терминального символа вводим нетерминал и дополнительную продукцию, а цепочку преобразуем по следующему правилу. Пусть исходная и результирующая цепочки имеют вид соответственно $Q = x_1 x_2 \ldots x_m$, $x_i \in V_T \cup V_N, i = 1,\overline{m}$, и $R = y_1 y_2 \ldots y_m$, $y_i \in V_N$, $i = 1,\overline{m}$. Элементы цепочек связаны следующим образом

$$y_i = \begin{cases} x_i, & x_i \in V_N, \\ X_{x_i}, & X_{x_i} \in V_N, x_i \in V_T, \\ P^* = P \cup \left\{X_{x_i}\right\} \end{cases}$$

В рассматриваемом примере, вторичные продукции, суть все продукции с правыми частями $a\mathbf{A}$ и $в\mathbf{B}$. Введением правил: $\mathbf{C} \to a$, $\mathbf{D} \to b$ эти продукции преобразуются к виду **CA** и **DB** соответственно. В результате имеем множество продукций P'

$$\mathbf{S} \to \mathbf{ABA}|\mathbf{CA}|\mathbf{DB}|a|b$$

$$\mathbf{A} \to \mathbf{CA}|\mathbf{DB}|a|b$$

$$\mathbf{B} \to \mathbf{DB}|b$$

$$\mathbf{C} \to a$$

$$\mathbf{D} \to b.$$

3. В расширенном множестве продукций Р' отыщем *третичные продукции*_в правых частях которых стоит *не менее трех нетерминалов подряд*, то есть, $R \to S_1S_2...S_m$, $S_i \in V_H$, $i = \overline{1,m}$, $m \geq 2$. Для каждой такой продукции строится разложение вида:

$$R \to \quad S_1N_{B1},$$
$$N_{B1} \to S_2N_{B2},$$
$$\dots$$
$$N_{Bm-1} \to S_{m-1}S_m.$$

Единственная, в рассмотренном примере, третичная продукция имеет вид: **S** $\to$ **ABA**. Заменим ее парой продукций **S** $\to$ **AE**, **E** $\to$ **BA.** Таким образом, нами получена грамматика в нормальной форме Хомского.

S→**AE**|**CA**|**DB**|*a*|*b*
A→**CA**|**DB**|*a*|*b*
B→**DB**|*b*
E→**BA**
C→*a*
D→*b*.

Привести грамматику к нормальной форме Грейбах.

Для построения грамматики, свободной от левой рекурсии в правилах, и обеспечения начала правых частей правил с терминальных символов, её приводят в нормальную форму Грейбах, продукции которой выглядят следующим образом:

$$A \to a\beta, A \in V_N,\ a \in V_T,\ \beta \in V^*.$$

Процесс образования базируется на теоремах [10], суть которых состоит в следующем.

Теорема 1. Если $\mathbf{A} \to \alpha\mathbf{B}\gamma$, **A**, **B** $\in V_N$, $\alpha,\gamma \in V^*$ правило КС – грамматики , а $\mathbf{B} \to \beta_1|\beta_2|\dots|\beta_m$ все продукции грамматики с нетерминалом **B** в левой части, то замена указанных продукций правилом $\mathbf{A} \to \alpha\beta_1\gamma|\alpha\beta_2\gamma|\dots|\alpha\beta_m\gamma$ - не отражается на языке порождаемом грамматикой.

Теорема 2. Для леворекурсивных продукций $\mathbf{A} \to \mathbf{A}\alpha_1|\mathbf{A}\alpha_2|\dots\mathbf{A}\alpha_m$ при остальных продукциях с нетерминалом **A** в левой части вида $\mathbf{A} \to \beta_1|\beta_2|\dots|\beta_k$ эквивалентной заменой будут продукции

$\mathbf{A} \to \beta_1|\beta_2|\dots|\beta_k|\beta_1\mathbf{C}|\beta_2\mathbf{C}\dots|\beta_k\mathbf{C}$.
$\mathbf{C} \to \alpha_1|\alpha_2|\dots|\alpha_m|\beta_1\mathbf{C}|\beta_2\mathbf{C}\dots|\beta_k\mathbf{C}$.

Пусть задана грамматика

S $\to$ **AB**|**A**
A $\to$ *a***A**|*b*|**B**
B $\to$ *b*|**B***b*.

На основании теоремы 2 избавимся от явной леворекурсивной продукции вида $\mathbf{B} \to b|\mathbf{B}b$. Имеем пару эквивалентных продукций $\mathbf{B} \to b\mathbf{C}|b$ и $\mathbf{C} \to b\mathbf{C}|b$.

Грамматика примет вид

$\mathbf{S}\to\mathbf{AB}|\mathbf{A}$
$\mathbf{A}\to a\mathbf{A}|b|\mathbf{B}$
$\mathbf{B}\to b|\mathbf{C}b$
$\mathbf{C}\to b\mathbf{C}|b$.

Последовательное применение теоремы 1 для правил с нетерминалами **S** и **A** в левой части даёт следующее.

$\mathbf{S}\to\mathbf{AB}$	$\Rightarrow$	$\mathbf{S}\to a\mathbf{AB}|b\mathbf{B}|\mathbf{BB}$,
$\mathbf{S}\to\mathbf{A}$	$\Rightarrow$	$\mathbf{S}\to a\mathbf{A}|b|\mathbf{B}$,
$\mathbf{S}\to\mathbf{BB}$	$\Rightarrow$	$\mathbf{S}\to b\mathbf{B}|b\mathbf{CB}$,
$\mathbf{S}\to\mathbf{B}$	$\Rightarrow$	$\mathbf{S}\to b|b\mathbf{C}$,
$\mathbf{A}\to\mathbf{B}$	$\Rightarrow$	$\mathbf{A}\to b|b\mathbf{C}$.

Окончательно, объединив правила, имеем

$\mathbf{S}\to a\mathbf{AB}|b\mathbf{B}|a\mathbf{A}|b\mathbf{CB}|b\mathbf{C}|b$
$\mathbf{A}\to a\mathbf{A}|b\mathbf{C}|b$
$\mathbf{B}\to b|b\mathbf{C}$
$\mathbf{C}\to b\mathbf{C}|b$.

Упражнения

1.1. Определить элементы грамматики: словари терминалов и нетерминалов, обозначить одну из возможных аксиом и язык, порождаемый грамматикой.

1.1.1. [5, с. 54]	$\mathbf{A}\to a|a\mathbf{AA}$.	1.1.2. [5, с. 37]	$\mathbf{A}\to a|a\mathbf{A}a$.
1.1.3.	$\mathbf{Z}\to a\mathbf{V}$ $\mathbf{V}\to a|\mathbf{W}$ $\mathbf{W}\to b\mathbf{W}|b$.	1.1.4.	$\mathbf{R}\to a\mathbf{Q}a$ $\mathbf{S}\to a\mathbf{Q}a|a$ $\mathbf{Q}\to b\mathbf{S}b|\mathbf{R}$.
1.1.5.	$\mathbf{S}\to a\mathbf{S}a$ $\mathbf{S}\to b\mathbf{S}b$ $\mathbf{S}\to c$.	1.1.6.	$\mathbf{A}\to a\mathbf{B}$ $\mathbf{A}\to b$ $\mathbf{B}\to \mathbf{A}a|b$.
1.1.7. [5, с. 77]	$\mathbf{S}\to a\mathbf{A}$ $\mathbf{A}\to a|\mathbf{B}$ $\mathbf{B}\to b\mathbf{B}|b$.	1.1.8. [5, с. 77]	$\mathbf{S}\to a\mathbf{B}a$ $\mathbf{A}\to a\mathbf{BA}|a$ $\mathbf{B}\to b\mathbf{A}b|b$.
1.1.9. [6, с.39]	$\mathbf{A}\to b\mathbf{A}|cc$.	1.1.10. [6, с.49]	$\mathbf{V}\to aa\mathbf{V}|bc$.
1.1.11.	$\mathbf{S}\to\mathbf{AB}$	1.1.12.	$\mathbf{S}\to\mathbf{A}c|\mathbf{B}d$

[10, с. 39] $\mathbf{A}\rightarrow\mathbf{X}|\mathbf{Y}$
$\mathbf{X}\rightarrow x|x\mathbf{X}$
$\mathbf{Y}\rightarrow y|y\mathbf{Y}$
$\mathbf{B}\rightarrow b|b\mathbf{B}$.

[3, с. 205] $\mathbf{A}\rightarrow a\mathbf{A}b|ab$
$\mathbf{B}\rightarrow a\mathbf{B}bb|abb$.

1.1.13. [3, с. 344] $\mathbf{S}\rightarrow\mathbf{AS}|a$
$\mathbf{A}\rightarrow b\mathbf{SA}|b$.

1.1.14. [3, с. 344] $\mathbf{S}\rightarrow\mathbf{SA}|\mathbf{A}$
$\mathbf{A}\rightarrow a\mathbf{A}|b$.

1.1.15. [3, с. 353] $\mathbf{S}\rightarrow\mathbf{AA}|\mathbf{AS}|b$
$\mathbf{A}\rightarrow\mathbf{SA}|\mathbf{AS}|a$.

1.1.16. [3, с. 448] $\mathbf{S}\rightarrow\mathbf{AB}$
$\mathbf{A}\rightarrow a$
$\mathbf{B}\rightarrow\mathbf{CD}|a\mathbf{E}$
$\mathbf{C}\rightarrow ab$
$\mathbf{D}\rightarrow\mathbf{B}b$
$\mathbf{E}\rightarrow bba$.

1.1.17. [3, с. 448] $\boldsymbol{S}\rightarrow 0\mathbf{S}1|\mathbf{A}$
$\mathbf{A}\rightarrow 1\mathbf{A}|1$.

1.1.18. [3, с. 448] $\mathbf{S}\rightarrow\mathbf{S}+\mathbf{A}|\mathbf{A}$
$\mathbf{A}\rightarrow a(\mathbf{S})|a|(\mathbf{S})$.

1.1.19. [3, с. 505] $\mathbf{S}\rightarrow a\mathbf{A}|\mathbf{B}$
$\mathbf{A}\rightarrow 0\mathbf{A}1|a$
$\mathbf{B}\rightarrow 0\mathbf{B}1|b$.

1.1.20. [3, с. 505] $\mathbf{S}\rightarrow a\mathbf{A}|b\mathbf{B}$
$\mathbf{A}\rightarrow 0\mathbf{A}1|01$
$\mathbf{B}\rightarrow 0\mathbf{B}1|01$.

1.1.21. [3, с. 505] $\mathbf{E}\rightarrow\mathbf{E}+\mathbf{T}|\mathbf{E}\text{-}\mathbf{T}|\mathbf{T}$
$\mathbf{T}\rightarrow\mathbf{T}*\mathbf{F}|\mathbf{T}/\mathbf{F}|\mathbf{F}$
$\mathbf{F}\rightarrow(\mathbf{E})|\text{-}\mathbf{E}|<\!iden\!>|<\!data\!>$.

1.1.22. [3, с. 516] $\mathbf{S}\rightarrow\mathbf{AB}|\mathbf{C}$
$\mathbf{A}\rightarrow a$
$\mathbf{B}\rightarrow\mathbf{CB}|\mathbf{A}$
$\mathbf{C}\rightarrow b$.

1.1.23. [4, с. 85] $\mathbf{S}\rightarrow\mathbf{A}|0\mathbf{B}$
$\mathbf{A}\rightarrow a\mathbf{A}b|0|0\mathbf{C}$
$\mathrm{B}\rightarrow a\mathbf{B}c|1|1\mathbf{C}$
$\mathbf{C}\rightarrow a\mathbf{C}|a$.

1.1.24. [4, с. 120] $\mathbf{S}\rightarrow a\mathbf{A}|a\mathbf{B}$
$\mathbf{A}\rightarrow b\mathbf{A}|c$
$\mathbf{B}\rightarrow b\mathbf{B}|d$.

1.1.25. [4, с. 190] $\mathbf{S}\rightarrow\mathbf{A}$
$\mathbf{A}\rightarrow a\mathbf{A}b\mathbf{A}c|a\mathbf{A}d|a$.

1.1.26. [3, с. 239] $\mathbf{E}\rightarrow\mathbf{E}+\mathbf{T}|\mathbf{T}$
$\mathbf{T}\rightarrow\mathbf{T}*\mathbf{F}|\mathbf{F}$
$\mathbf{F}\rightarrow(\mathbf{E})|sin(\mathbf{E})|cos(\mathbf{E})$
$\mathbf{F}\rightarrow<\!iden\!>|<\!data\!>$.

1.1.27. [11, с. 206] $\mathbf{S}\rightarrow x\mathbf{S}y|a|b$.

1.1.28. 5.2.13.

1.1.29. 5.2.11.

1.1.30. 5.2.12.

1.2. Построить контекстно-свободные грамматики, порождающие следующие предложения языка.

1.2.1. $L=\{a^{3n},\ n\geq 1\}$;

1.2.2. $L=\{a^{n}b^{2m-1},\ n,m\geq 1\}$;

1.2.3. $L=\{a^{n}b^{n},\ n\geq 1\}$;

1.2.4. $L=\{a^{n}b^{m}c^{n},\ n,m\geq 1\}$;

1.2.5. [5, с. 78] $L=\{a^{n}b^{n},\ n\geq 2\}$;

1.2.6. [5, с. 78] $L=\{(ab)^{n},\ n\geq 2\}$;

1.2.7. $L=\{a^m b, m\geq 1\}$;
[5, с.153]

1.2.8. $L=\{c^n b, a^n b, n\geq 1\}$;
[5, с.154]

1.2.9. $L=\{ac^m d, bc^m d, m\geq 1\}$;
[5, с.162]

1.2.10. $L=\{a^m b^n c^p, m,n,p\geq 0\}$;
[11, с. 39]

1.2.11. $L=\{a^3 b^3\}$;
[10, с. 31]

1.2.12. $L=\{ab^n a, n\geq 1\}$;
[6, с. 39]

1.2.13. $L=\{a^m b^m c^p, m,p\geq 0\}$;
[11, с. 39]

1.2.14. $L=\{x^n a y^n, x^n b y^n, n\geq 0\}$;
[11, с. 39]

1.2.15. $L=\{a^n b^n c, a^n b^{2n} d, n\geq 1\}$;
[3, с. 205]

1.2.16. $L=\{a^n 0 a^i b^n, i,n\geq 0;\ 0a^n 1 a^i c^n, i,n\geq 0\}$
[4, с. 85]

1.2.17. $L=\{a^n b^n, a^n c^n, n\geq 1\}$;
[4, с. 162]

1.2.18. $L=\{a^n a^{n^3}, n\geq 1\}$;
[4, с. 261]

1.2.19. $L=\{a^n a^{2^n}, n\geq 1\}$.
[4, с. 216]

1.3. Какая из грамматик является грамматикой фазовой структуры, а какая нет и почему?

1.3.1. [5, с. 14]
A→*aba*
AB→*b*
B→*b*A.

1.3.2. [5, с. 14]
A→*aba*
AB→*b*
b→*b*A
A→*ε*.

1.3.3. [3, с. 218]
S→*a***S***b*|*ε*.

1.3.4. [3, с. 218]
S→**AS**|*b*
A→**SA**|*a*.

1.4. Проанализируйте данные грамматики с позиции контекстной зависимости или бесконтекстности грамматики.

1.4.1. [5, с. 51]
A→**B**
B→*ab***A**.

1.4.2. [5, с. 51]
A→**AB**
AB→AG
G→*ab***G**.

1.4.3. [5, с. 51]
A→**AB**
AB→**BG**
B→*a*.

1.4.4. [9, с. 43]
I→|**LI**
L→*a*|*b*|*c*.

1.4.5. [9, с. 44]
A→*a***B**
B→*a***B**|*b*.

1.4.6. [9, с. 45]
A→*a***AB**|*abc*
*b***B**→*bbc*
*c***B**→**B***c*.

1.4.7. [9, с. 45]
A→*a***A***a*|*c*
A→*b***A***b*.

1.4.8. [9, с. 45]
A→*a***A***a*|*aa*
A→*b***AB**
B→ *bb*.

1.5. Классифицируйте данные грамматики: являются ли они грамматиками фазовой структуры (ГФС) и, если являются, то контекстно-свободными или контекстно-зависимыми? [5, с. 55]

1.5.2.
A→*ε*|**B**
BA→*a*
ε→**B**.

1.5.2.
a→**A**
A→B
B→*a*.

1.5.3.
A→*a*|**B**
B→*b*.

1.6. Пусть правила грамматики имеют вид [6, с. 53]

$\mathbf{S}\rightarrow a\mathbf{P}$

$\mathrm{P}\rightarrow \mathrm{A}|b$

$\mathbf{A}\rightarrow c.$

Проверьте истинность или ложность следующих утверждений:

$\mathbf{S}\Rightarrow a,$	$a\mathbf{P}\Rightarrow^{+} ac,$
$\mathbf{S}\Rightarrow^{*} a,$	$\mathbf{S}\Rightarrow ac,$
$\mathbf{P}\Rightarrow^{*} \mathbf{A}c,$	$\mathbf{S}\Rightarrow^{+} ac,$
$a\mathbf{S}\Rightarrow^{*} aab,$	$\mathbf{S}\Rightarrow^{+} ab,$
$a\mathbf{P}b\Rightarrow acb,$	$\mathbf{S}\Rightarrow^{*} ab,$
$\mathbf{S}\Rightarrow^{*} a\mathbf{A},$	$a\mathbf{A}\Rightarrow ac.$

1.7. Построить синтаксические деревья для цепочек грамматики [5, с. 78]

Грамматика	Цепочки
$\mathbf{S}\rightarrow a\mathbf{B}\vert b\mathbf{A}$	*aaabbb,*
$\mathbf{A}\rightarrow a\vert a\mathbf{S}\vert b\mathbf{AA}$	*aababb,*
$\mathbf{B}\rightarrow b\vert b\mathbf{S}\vert a\mathbf{BB}.$	*abababba,*
	abaababb,
	bababbbaa,
	bbbaababaaba.

1.8. Для грамматики $\mathbf{V}\rightarrow a|b|c|\mathbf{V}a|\mathbf{V}b|\mathbf{V}c$ выведите, когда можно, цепочки: *a, ab0, a0c0, 0a,111aaa* [6, с.39].

1.9. Построить грамматику:

1.9.1. язык которой состоит из чётных целых чисел [6, с.39];

1.9.2. язык которой представляют целые десятичные числа: положительные, отрицательные и нуль [5, с. 55];

1.9.3. язык которой определён на множестве $\{0,1\}$, и непосредственно справа от *0* стоит *1* [10, с. 37];

1.9.4. язык которой определён на множестве $\{0,1\}$, и строка является палиндромом (читается одинаково в прямом и обратном порядке) [10, с. 37];

1.9.5. язык которой определён на множестве $\{0,1\}$, и нулей в строке вдвое больше чем единиц [10, с. 37];

1.9.6. язык которой определён на множестве $\{0,1\}$ и включает все возможные строки, включая и пустую [11, с. 206].

1.10. Пользуясь круговой диаграммой, описывающей элемент языка программирования высокого уровня Pascal [7], построить КС-грамматику

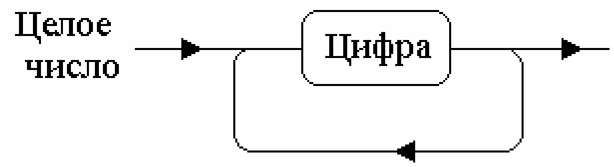

Рисунок 1.3 – Синтаксис целого числа без знака

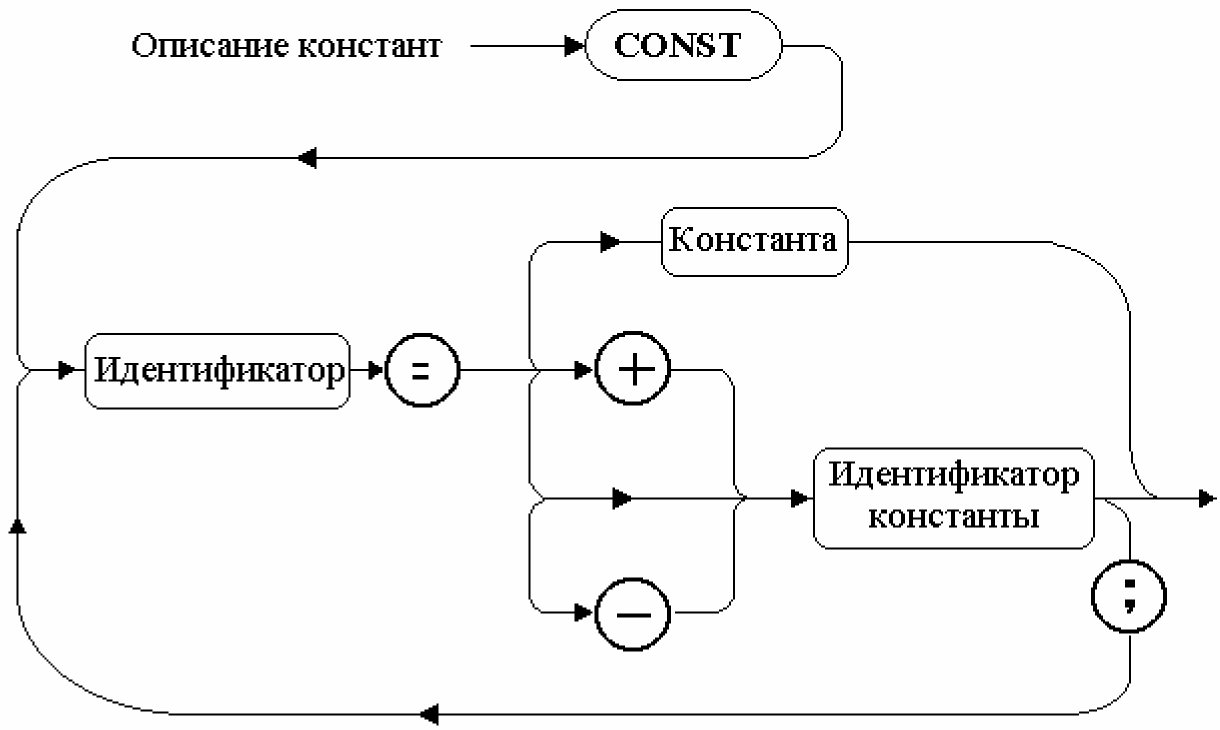

Рисунок 1.4 – Синтаксис определения констант

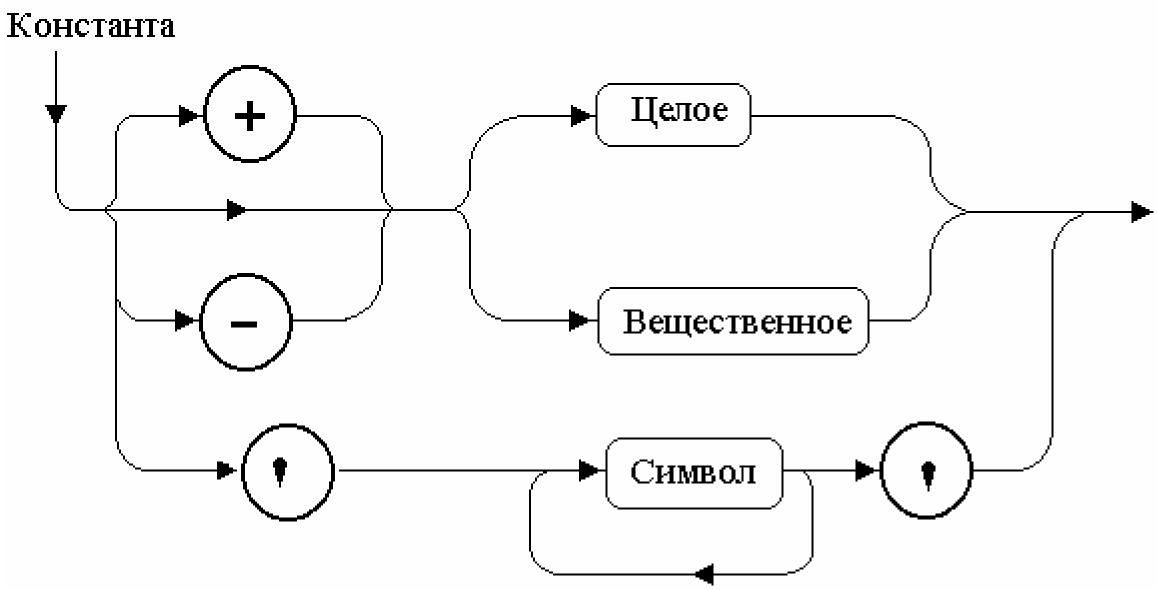

Рисунок 1.5 – Синтаксис константы

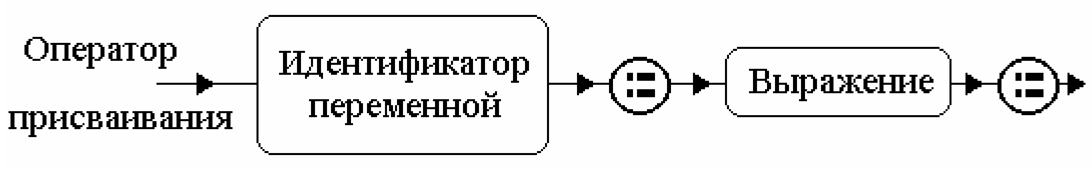

Рисунок 1.6 – Синтаксис оператора присваивания

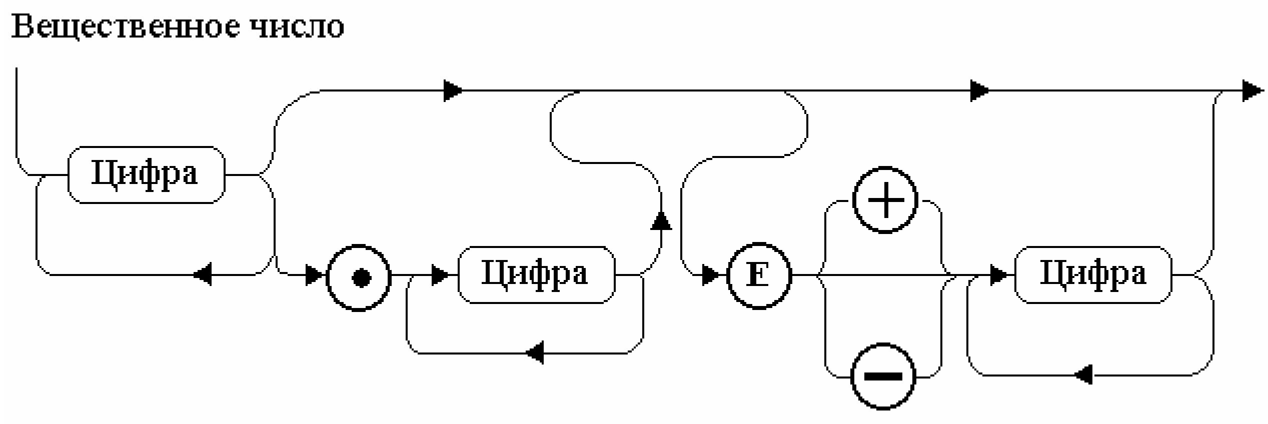

Рисунок 1.7 – Синтаксис вещественного числа

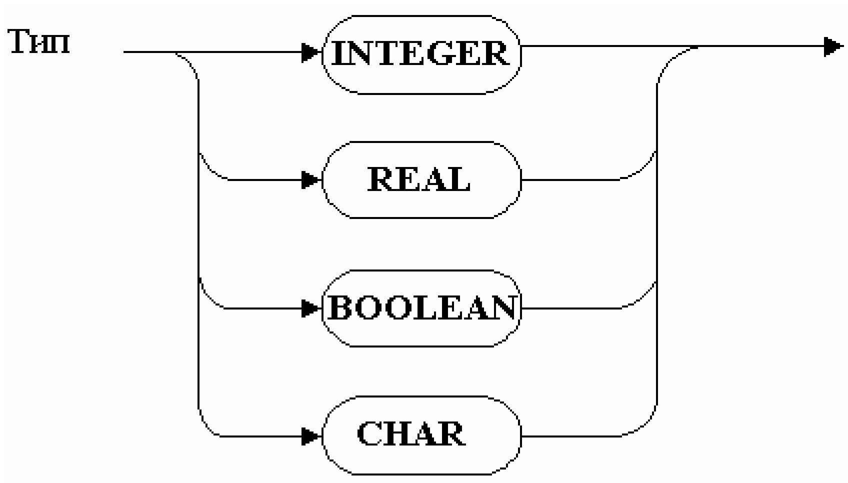

Рисунок 1.8 – Синтаксис типа

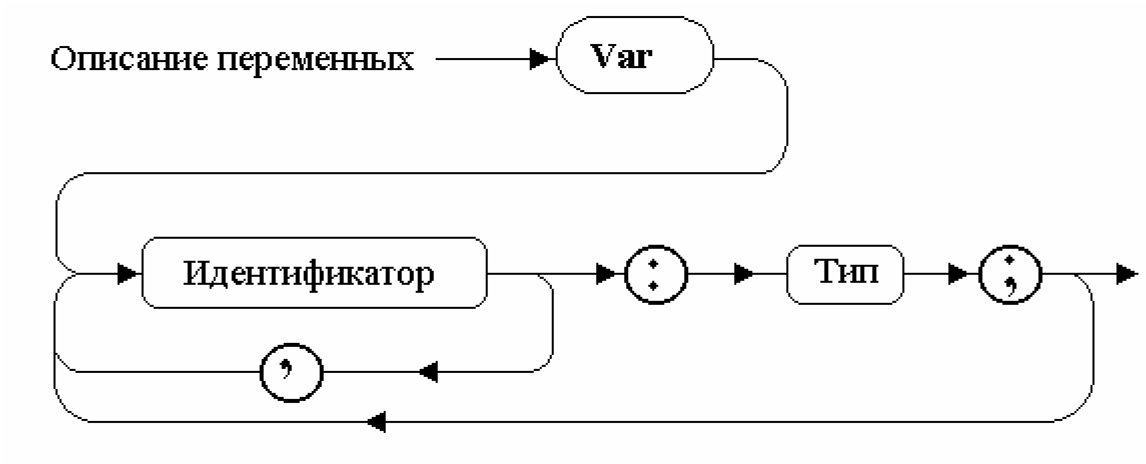

Рисунок 1.9 – Синтаксис описания переменной

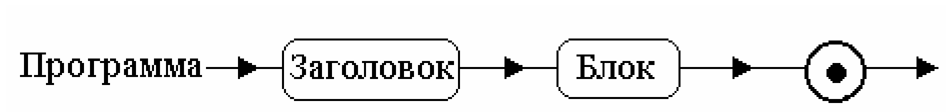

Рисунок 1.10 – Синтаксис программы

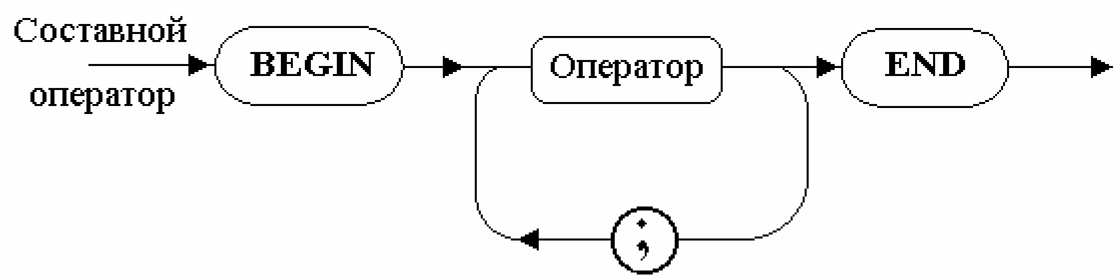

Рисунок 1.11 – Синтаксис составного оператора

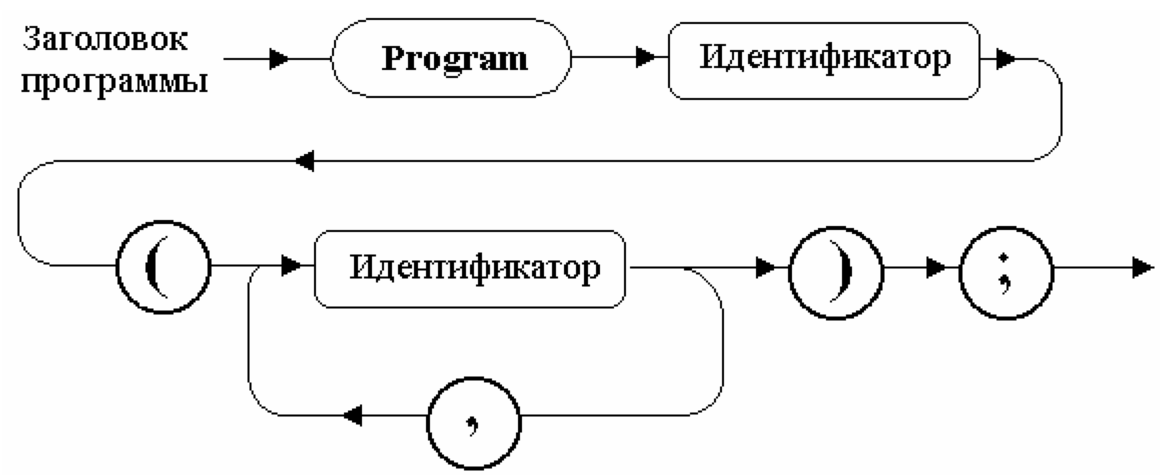

Рисунок 1.12 – Синтаксис заголовка

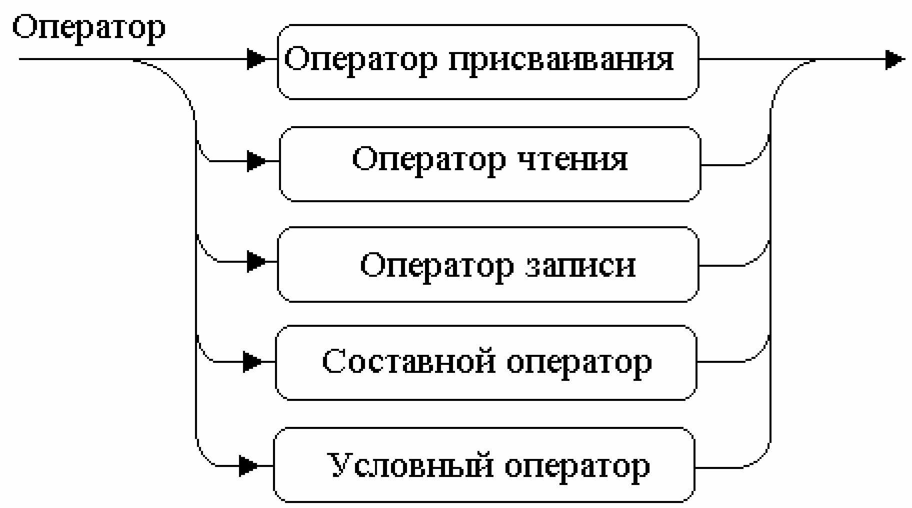

Рисунок 1.13 – Синтаксис оператора

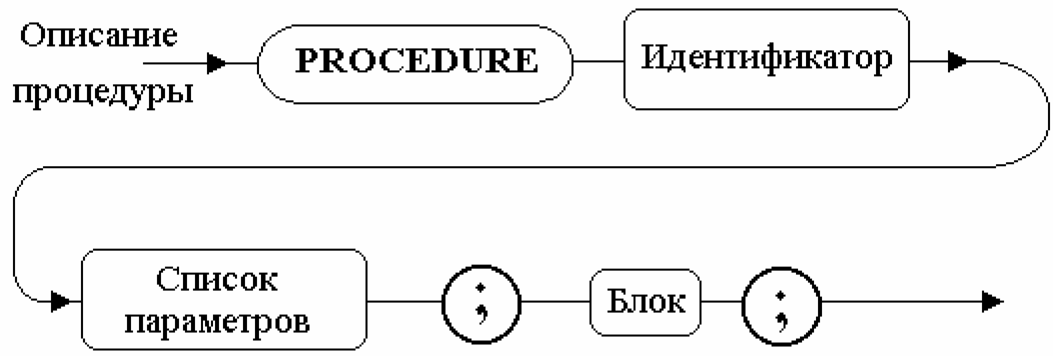

Рисунок 1.14 – Синтаксис описания процедуры

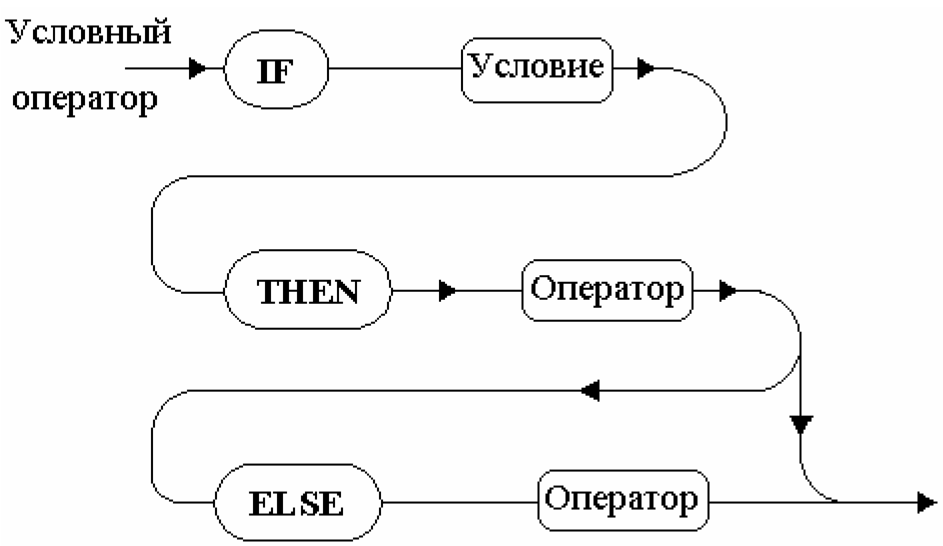

Рисунок 1.15 – Синтаксис условного оператора

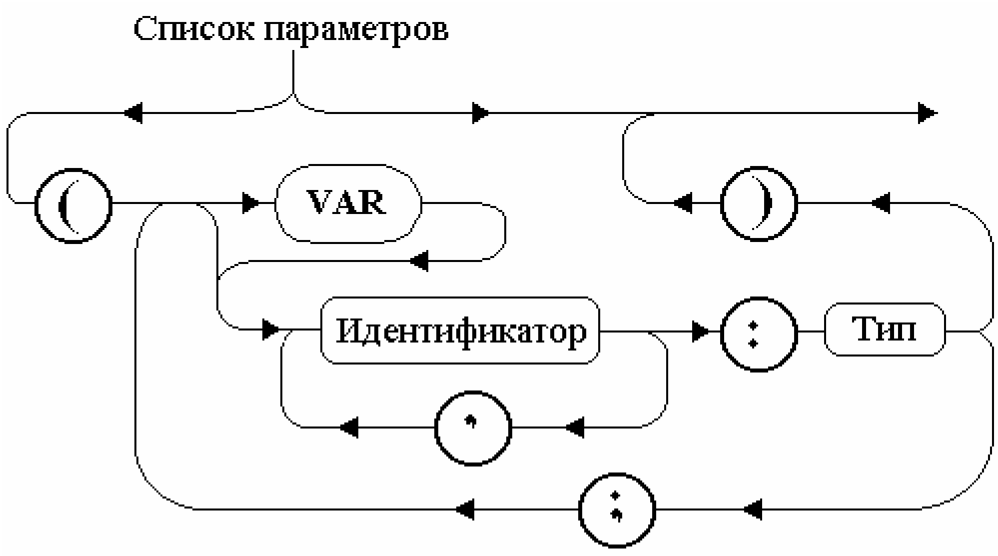

Рисунок 1.16 – Синтаксис списка параметров

Рисунок 1.17 – Синтаксис блока

1.11. Являются ли нижеприведенные пары грамматик однозначными и эквивалентными?

1.11.1.	$\mathbf{S}\to\mathbf{A}b\vert a\mathbf{B}$	и	$\mathbf{S}\to\mathbf{A}b$
[5, с. 77]	$\mathbf{A}\to a$		$\mathbf{A}\to a.$
	$\mathbf{B}\to b$		
1.11.2.	$\mathbf{S}\to\mathbf{A}$	и	$\mathbf{S}\to\mathbf{AB}$
[5, с. 77]	$\mathbf{A}\to a\vert a\mathbf{A}a$		$\mathbf{A}\to ab\vert a$
			$\mathbf{B}\to bc\vert c.$
1.11.3.	5.2.4.	и	5.2.5..
1.11.4.	$\mathbf{S}\to\mathbf{AB}$	и	$\mathbf{S}\to a\mathbf{B}\vert b\mathbf{AS}\vert b\mathbf{A}$
[10, с. 38]	$\mathbf{A}\to a\vert a\mathbf{S}\vert b\mathbf{AA}$		$\mathbf{A}\to b\mathbf{AA}\vert a$
	$\mathbf{B}\to b\vert b\mathbf{S}\vert a\mathbf{BB}$		$\mathbf{B}\to a\mathbf{BB}\vert b.$
1.11.5.	$\mathbf{S}\to\mathbf{AB}$	и	$\mathbf{Z}\to\mathbf{XB}\vert\mathbf{Y}$

[4, с. 148] $\mathbf{A}\to a\mathbf{A}|\varepsilon$ $\quad$ $\mathbf{X}\to a\mathbf{A}$

$\mathbf{B}\to b\mathbf{A}|\varepsilon$ $\quad$ $\mathbf{A}\to a\mathbf{A}|\varepsilon$

$\mathbf{Y}\to b\mathbf{A}$

$\mathbf{B}\to b\mathbf{A}|\varepsilon$.

1.11.7. $\mathbf{A}\to a\mathbf{C}|b\mathbf{B}$ и $\mathbf{A}\to a\mathbf{C}|a\mathbf{CA}|b\mathbf{B}|b\mathbf{BA}$

[9, с. 56] $\mathbf{B}\to a|a\mathbf{A}|b\mathbf{BB}$ $\quad$ $\mathbf{B}\to a|b\mathbf{BB}$

$\mathbf{C}\to b|b\mathbf{A}|aac$ $\quad$ $\mathbf{C}\to b|a\mathbf{CC}$.

1.12. Показать, что для цепочек <*iden*>+<*iden*>*<*iden*> и <*iden*>+<*iden*>+<*iden*> нижеследующая грамматика неоднозначна [6, с.46].

V→<*iden*>|(**V**)|**V**<**Kod_Op**>**V**

<**Kod_Op**>→ + | - | * | /

1.13. Однозначна ли нижеследующая грамматика?

1.13.1. $\mathbf{E}\to\mathbf{T}|\mathbf{E}+\mathbf{T}|\mathbf{E}-\mathbf{T}$

[9, с. 33] $\mathbf{T}\to\mathbf{F}|\mathbf{T}*\mathbf{F}|\mathbf{T}/\mathbf{F}$

$\mathbf{F}\to a|b|c|(\mathbf{E})$;

1.13.2. **S**→*IF b THEN* **S** *ELSE* **S**

[3, с. 231] **S**→*IF b THEN* **S**|*a*;

1.13.3. **S**→*IF b THEN* **S**| *IF b THEN* **T** *ELSE* **S**|*a*

[3, с. 233] **T**→*IF b THEN* **T** *ELSE* **T**|*a*;

1.13.4. $\mathbf{S}\to\mathbf{AB}|\mathbf{DC}$

[3, с. 233] $\mathbf{A}\to a\mathbf{A}|\varepsilon$

$\mathbf{B}\to b\mathbf{B}c|\varepsilon$

$\mathbf{C}\to c\mathbf{C}|\varepsilon$

$\mathbf{D}\to d\mathbf{D}b|\varepsilon$;

1.13.5. $\mathbf{E}\to\mathbf{E}+\mathbf{E}|\mathbf{E}*\mathbf{E}$

[3, с. 494] **E**→(**E**)|<*iden*>;

1.13.6. $\mathbf{E}\to\mathbf{B}|\mathbf{U}$

[9, с. 54] $\mathbf{B}\to(\mathbf{E}+\mathbf{E})$

$\mathbf{U}\to \mathrm{sin}\mathbf{E}|-|\mathrm{sqrt}\mathbf{E}$.

1.14. Какие цепочки порождает контекстно - зависимая грамматика [10, с. 38]? Как Вы думаете, можно ли построить эквивалентную ей контекстно - свободную грамматику?

$\mathbf{E}\to a\mathbf{SBC}|a\mathbf{BC}$

$\mathbf{CB}\to\mathbf{BC}$

$a\mathbf{B}\to ab$

$b\mathbf{B}\to bb$

$b\mathbf{C}\to bc$

$c\mathbf{C}\to cc$.

1.15. Разработать грамматику и построить синтаксические деревья для операторов языка [4, с. 205].

1.15.1. `A:=(B-C)/(B+C).`

1.15.2. `I:=length(X!Y).`

```
if B>C then
```

1.15.3.
```
if B>C then
        if D>E then A:=B+C
             else A:=B-C
    else A:=B*C.
```

Экивалентные преобразования грамматик

1.16. Построить *ε*-свободную грамматику, эквивалентную заданной

1.16.1. [5, с. 78]	**S**→**A** **A**→*a***A**|*a*|*ε*.	1.16.2. [5, с. 78]	**S**→**AB** **A**→**G**|*ab* **G**→*c*|*ε* **B**→*a***A***b*|*ε*.
1.16.3.	**S**→*ε*|*a***C**|**A***b* **C**→*a***C**|*a***B**|*ε* **B**→*b***B**|*b*|*ε*.	1.16.4. [10, с. 37]	**S**→**AB** **A**→**SA**|**BB**|*b***B** **B**→*b*|*a***A**|*ε*.
1.16.5. [3, с. 173]	**S**→*a***S***b***S**|*b***S***a***S**|*ε*	1.16.6.	2.3.4.
1.16.7. [3, с. 133]	**S**→*0***A**|*1***S**|*ε* **A**→*0***B**|*1***A** **B**→*0***S**|*1***B**.	1.16.8. [3, с. 189]	**S**→**ABC** **A**→**BB**|*ε* **B**→**CC**|*a* **C**→**AA**|*b*.
1.16.9. [3, с. 189]	**S**→**A**|**B** **A**→**C**|**D** **B**→**D**|**E** **C**→**S**|*a*|*ε* **D**→**S**|*b* **E**→**S**|**C**|*ε*.	1.16.10. [3, с. 313]	**S**→**AS**|**BB** **A**→*b***AA**|*a* **B**→*b*|*ε*.
1.16.11. [3, с. 430]	**S**→**A***b*|**B***c* **A**→**A***a*|*ε* **B**→**B***a*|*ε*.	1.16.12. [3, с. 434]	**S**→**AB** **A**→**B***a*|*ε* **B**→**C***b*|*ε* **C**→*c*|*ε*.
1.16.13. [3, с. 448]	**S**→**AB** **A**→*0***A***1*|*ε* **B**→*1***B**|*1*.	1.16.14. [4, с. 131]	**S**→**AB** **A**→*a***B***b*|*ε* **B**→*b***B**|*b*.
1.16.15. [10, с. 78]	**S**→*a***A**|*a***BB** **A**→**B***a*|**S***b* **B**→*b***AS**|*ε*.	1.16.16. [4, с. 137]	**S**→*IF b THEN* **SE**|*a* **E**→*ELSE* **S**|*ε*.
1.16.17.	E→**TW**	1.16.18.	**S**→**ABC**

[4, с. 162] $\mathbf{W}\to+\mathbf{TW}|\varepsilon$
$\mathbf{T}\to\mathbf{FU}$
$\mathbf{U}\to*\mathbf{FU}$
$\mathbf{F}\to a|(\mathbf{E})$.

[11, с. 206] $\mathbf{A}\to a\mathbf{A}|\varepsilon$
$\mathbf{B}\to b\mathbf{B}|\varepsilon$
$\mathbf{C}\to c\mathbf{C}|\varepsilon$.

1.16.19. [11, с. 206]
$\mathbf{S}\to\mathbf{TC}$
$\mathbf{T}\to a\mathbf{T}b|\varepsilon$
$\mathbf{C}\to c\mathbf{C}|\varepsilon$.

1.16.20. [11, с. 212]
$\mathbf{S}\to 0\mathbf{S}|1\mathbf{T}|\varepsilon$
$\mathbf{T}\to 0\mathbf{S}|\varepsilon$.

1.16.21 [11, с. 212]
$\mathbf{S}\to 0\mathbf{U}|1\mathbf{T}$
$\mathbf{T}\to 0\mathbf{X}|1\mathbf{UU}$
$\mathbf{U}\to 1\mathbf{X}|0\mathbf{TT}$
$\mathbf{X}\to\mathbf{S}|\varepsilon$.

1.16.22. Пример 1.9.6..

1.16.23. Пример 1.13.4..

1.16.24. Пример 1.13.1.6.

1.17. Привести грамматику к нормальной форме Хомского

1.17.1.
$\langle\mathbf{T}\rangle \to \langle\mathbf{P}\rangle \mid \hat{}\langle\mathrm{iden}\rangle \mid \langle\mathbf{OPIS}\rangle \mid \mathit{PACKED}\ \langle\mathbf{OPIS}\rangle$
$\langle\mathbf{OPIS}\rangle \to \mathit{SET\ OF}\ \langle\mathbf{P}\rangle \mid \mathit{ARRAY}\ [\langle\mathbf{DIM}\rangle]\ \mathit{SET\ OF}\ \langle\mathbf{T}\rangle$
$\langle\mathbf{P}\rangle \to \mathit{REAL} \mid \mathit{CHAR} \mid \mathit{INTEGER}$
$\langle\mathbf{DIM}\rangle \to \langle\mathbf{P}\rangle \mid \langle\mathbf{P}\rangle, \langle\mathbf{P}\rangle \mid \langle\mathbf{P}\rangle, \langle\mathbf{P}\rangle, \langle\mathbf{P}\rangle$.

1.17.2.
$\mathbf{Z}\to\mathbf{X}|0\mathbf{XYX}1$
$\mathbf{X}\to 0|0\mathbf{X}|0\mathbf{YX}$
$\mathbf{Y}\to 1|1\mathbf{Y}|1\mathbf{X}$.

1.17.3. [3, с. 177]
$\mathbf{S}\to a\mathbf{AB}|\mathbf{BA}$
$\mathbf{A}\to\mathbf{BBB}|a$
$\mathbf{B}\to\mathbf{AS}|b$.

1.17.4. [3, с. 180]
$\mathbf{S}\to 0\mathbf{S}1|01$.

1.17.5. [3, с. 180]
$\mathbf{S}\to a\mathbf{B}|b\mathbf{A}$
$\mathbf{A}\to a\mathbf{S}|b\mathbf{AA}|a$
$\mathbf{B}\to b\mathbf{S}|a\mathbf{BB}|b$.

1.17.6. [3, с. 218]
$\mathbf{S}\to\mathbf{SS}|\mathbf{A}$
$\mathbf{A}\to 0\mathbf{A}1|\mathbf{S}|01$.

1.17.7.
$\mathbf{S}\to a\mathbf{A}|a\mathbf{BB}$
$\mathbf{A}\to\mathbf{B}a|\mathbf{S}b$
$\mathbf{B}\to b\mathbf{AS}|a$.

1.17.8. [4, с. 145]
$\mathbf{A}\to\mathbf{A}|\mathbf{B}$
$\mathbf{A}\to a\mathbf{A}b|0$
$\mathbf{B}\to a\mathbf{B}bb|1$.

1.17.9. [4, с. 146]
$\mathbf{S}\to\mathbf{A}|\mathbf{B}$
$\mathbf{A}\to aa\mathbf{A}|aa$
$\mathbf{B}\to aa\mathbf{B}|a$.

1.17.10. [4, с. 146]
$\mathbf{S}\to\mathbf{A}b|\mathbf{A}c$
$\mathbf{A}\to\mathbf{AB}|a$
$\mathbf{B}\to a$.

1.17.11. [4, с. 146]
$\mathbf{S}\to\mathbf{A}b|\mathbf{B}c$
$\mathbf{A}\to\mathbf{A}a|a$
$\mathbf{B}\to\mathbf{B}a|a$.

1.17.12. [4, с. 146]
$\mathbf{S}\to\mathbf{A}b|\mathbf{B}c$
$\mathbf{A}\to\mathbf{A}c|a$
$\mathbf{B}\to\mathbf{B}c|a$.

1.17.13. [4, с. 146]
$\mathbf{S}\to\mathbf{BA}b|\mathbf{CA}c$
$\mathbf{A}\to\mathbf{BA}|a$
$\mathbf{B}\to a$
$\mathbf{C}\to c$.

1.18. Привести грамматику к нормальной форме Грейбах

1.18.1.
$\mathbf{S}\to\mathbf{AS}|\mathbf{AB}$
$\mathbf{A}\to\mathbf{BS}|a$

1.18.2.
$\mathbf{A}\to\mathbf{BC}$
$\mathbf{B}\to\mathbf{CA}|\mathbf{B}$

$\mathbf{B} \to \mathbf{AA}|b.$

$\mathbf{C} \to \mathbf{A}b|a.$

1.18.3. [10, с. 78]
$\mathbf{S} \to \mathbf{AB}$
$\mathbf{A} \to \mathbf{BS}|b$
$\mathbf{B} \to \mathbf{SA}|a.$

1.18.4. [3, с. 108]
$\mathbf{E} \to \mathbf{E}+\mathbf{T}|\mathbf{T}$
$\mathbf{T} \to \mathbf{T}*\mathbf{F}|\mathbf{F}$
$\mathbf{F} \to (\mathbf{E})|<\mathit{iden}>.$

1.18.5. [3, с. 205]
$\mathbf{S} \to \mathbf{B}a|\mathbf{A}b.$
$\mathbf{A} \to \mathbf{S}a|\mathbf{AA}b|a$
$\mathbf{B} \to \mathbf{S}b|\mathbf{BB}a|b.$

1.19. Определить грамматику в нормальной форме Грейбах, порождающую арифметические выражения над словарем $\{a, b, c\}$

2. ПОСТРОЕНИЕ КОНЕЧНЫХ АВТОМАТОВ

Цели и задачи раздела:

а) научиться производить построение детерминированных конечных автоматов (ДКА), допускающих определённые цепочки символов языка;

б) освоить приёмов описания конечных автоматов (КА) в виде графов, таблиц переходов и регулярных выражений;

в) научиться производить построение формальной автоматной грамматики, соответствующей конечному автомату и наоборот;

г) выполнять построения ДКА по недетерминированным конечным автоматам (НКА);

д) проводить построение минимальных детерминированных конечных автоматов (МДКА).

Система ссылок на теоретические источники

Теоретические положения, подлежащие усвоению	Исчерпывающее изложение	Доходчивое изложение	Изложение, адаптированное к восприятию
1. Конечные автоматы и их элементы	[6, 8, 9, 10]	[6, 9, 12]	[2, с. 6]
2. Методы описания КА	[6, 8, 9, 10]	[6, 9, 12]	[2, с. 9 - 10]
3. Система соответствия формальных грамматик и ДКА	[6, 8, 9, 10, 12]	[6, 9, 12]	[2, с. 9]
4. Принципы построение КА, допускающего определённые цепочки символов	[6, 8, 9, 12]	[6, 9, 12]	[2, с. 12 - 15]
5. Программирование КА	[6, 8, 9]	[6, 9]	[2, с. 14 - 15]
6. Описание КА регулярными выражениями	[6, 8, 9]	[6, 9]	[2, с. 15 - 16]
7. Построение ДКА по НКА	[6, 8, 9]	[6, 9]	[2, с. 10 - 11]
8. Методы минимизации КА	[6, 8, 9]	[9]	[2, с. 19 - 23]

Примеры выполнения заданий

Построить минимальный детерминированный конечный автомат (ДКА), допускающий определённые цепочки символов языка, и формальную автоматную грамматику, соответствующую конечному автомату.

Выполним это упражнение для КА, который будет принимать любое английское слово, начинающееся на ***un*** и заканчивающееся на ***d*** [2, с.14], и, таким образом, разбивать все слова языка на два множества.

Построим регулярное выражение по правилам [2, с. 15 - 17, 10]:

$$un\{u\vee n\vee\sigma\vee d\}d\{d\}\rho,$$

где ***u***, ***n***, ***d***, - искомые символы, σ - любой символ, ρ - пробел. Это выражение подвергнем минимизации по алгоритму профессора Е. А. Бутакова. Расставим индексы мест регулярного выражения. *Место регулярного выражения* - это промежуток между двумя буквами, буквой и знаком, буквой и скобкой, а также начало и конец выражения. Различают следующие *типы мест*: а) начало выражения - *начальное место*, б) конец выражения - *конечное место*, в) *основное место* – место, слева от которого стоит буква, а также начальное место, г) *предосновное место* - место, справа от которого стоит буква.

Сперва выставляют индексы основных мест в порядке возрастания (короткие линии на рисунке), затем индексы предосновных мест (длинные линии на рисунке). При выставлении индексов предосновных мест руководствуются правилами. а) Индекс перед любыми скобками распространяется на все начальные места дизъюнктивных членов, записанных в этих скобках. б) Индекс конечного места любого дизъюнктивного члена, заключённого в любые скобки, распространяется на место, непосредственно следующее за этими скобками. в) Индекс места перед итерационными скобками распространяется на место, непосредственно следующее за этими скобками. г) Индекс места за итерационными скобками распространяется в начальные места всех дизъюнктивных членов в этих скобках. д) Индекс конечного места любого дизъюнктивного члена, заключённого в итерационные скобки, распространяется на начальные места всех дизъюнктивных членов, заключённых в эти скобки. е) Индексы места, справа и слева от которого стоят буквы никуда не распространяется. ж) Индекс конечного места распространяется на те же места, на которые и индекс начального места. Расстановка индексов представлена на рисунке 2.1.

Правило минимизации: Если несколько предосновных мест отмечено одинаковой совокупностью индексов, и справа от этих мест записаны одинаковые буквы, то основные места, расположенные справа от этих букв, можно отметить одинаковыми индексами. Под это правило подпадают позиции с номерами 6 и 7. Переопределяем индексы, и выполняем разметку. Новая разметка представлена на рисунке 2.2.

Дальнейшую минимизацию по регулярному выражению произвести не удаётся.

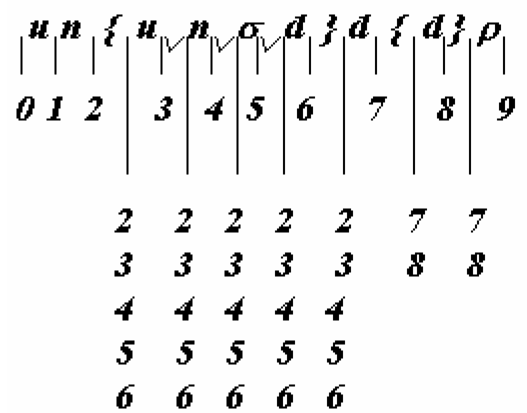

Рисунок 2.1 - Первоначальная разметка регулярного выражения

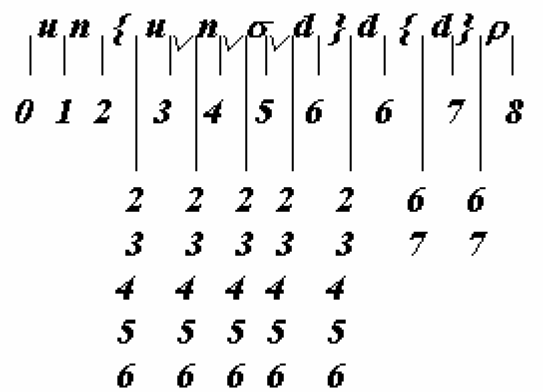

Рисунок 2.2 - Разметка регулярного выражения после минимизации

Анализ разметки выражения показывает, что автомат является *недетерминированным*, ибо образ функции перехода δ(6, ***d***)={6,7} определён неоднозначно. Выполним построение ДКА по НДКА. Алгоритм построения состоит в следующем.

1) Если НКА является частичным, то доопределим функцию перехода, вводя фиктивное состояние. 2) Построим таблицу переходов и, с её помощью, рассмотрим функцию переходов. Если значение функции не содержится в множестве состояний автомата, то дополним это множество данной функцией. 3) Для вновь введенного состояния строим функцию переходов; анализируем функцию переходов в состояниях, определяющих данное состояние. Определяем выходной сигнал, как дизъюнкцию выходных сигналов, входящих в данное обобщенное состояние состояний. 4) Повторяем п.п. 2 - 3 до тех пор, пока множество состояний автомата не будет изменено.

Введём обобщённое состояние {6,7} для переходов 6 → 6 и 6 → 7, проанализируем его по входной строке. Иллюстрация применения алгоритма построения детерминированного конечного автомата для данного конкретного случая, приводится на рисунке 2.3.

Обозначим обобщённое состояние {6, 7} ≡ 7, введём состояния: 9 с выходным сигналом *Error*, в котором автомат будет дожидаться пробела (ρ), означающего окончание слова, 10 с сигналом *Bed*, соответствующего слову, не удовлетворяющему условию задачи, сигнал *Good* для состояния 8 («правильное» слово), сигнал *x* для всех рабочих состояний. Дополним все неопределенные образы функции переходов переходами в состояние 9, а частично определенные состояния {3, –}, {4, –}, {5, –} дополним до {3, 3}, {4, 4}, {5, 5}. Построим таблицу переходов и выходов конечного автомата.

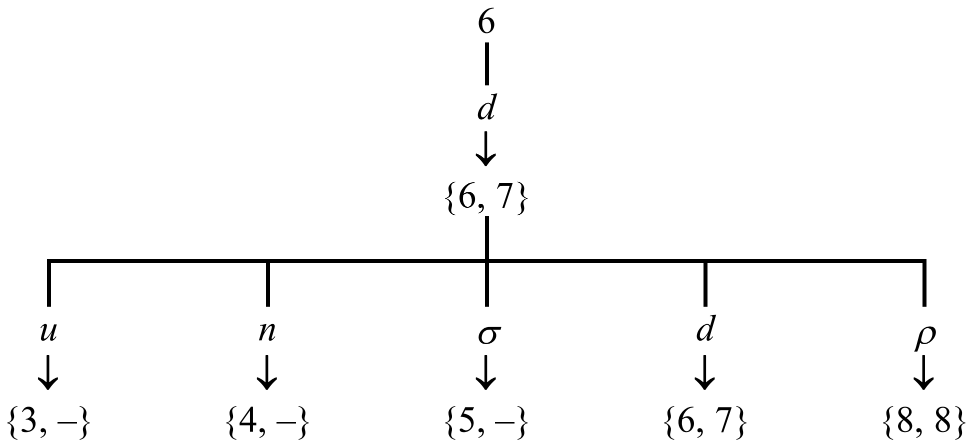

Рисунок 2.3 – Приведение НКА к ДКА

Таблица 2.1 - Функция переходов и выходов ДКА

	Сигнал	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*Good*	*Error*	*Bed*
	Состояние	0	1	2	3	4	5	6	7	8	9	10
Вход	***u***	1	9	3	3	3	3	3	3	-	9	-
	n	9	2	4	4	4	4	4	4	-	9	-
	d	9	9	6	6	6	6	7	7	-	9	-
	$\boldsymbol{\rho}$	10	10	10	10	10	10	8	8	-	10	-
	$\boldsymbol{\sigma}$	9	9	5	5	5	5	5	5	-	9	-

Из таблицы видно, группы столбцов 2 – 5 и 6,7 имеют одинаковое содержимое и выходные сигналы. Следовательно, *они не различимы по входной строке и по выходному сигналу,* и автомат можно минимизировать по таблице. Обозначим: 0→0 (S), 1→1(A), {2 – 5}→2 (B), {6,7}→3 (C), 8→6 (F), 9→4 (D), 10→5 (E). Преобразуем функцию переходов и выходов КА. Функция переходов МДКА приводится в таблице 2.2, а соответствующий ему граф показан на рисунке 2.4.

Таблица 2.2 – Функция переходов и выходов минимального ДКА

	Сигнал	*x*	*x*	*x*	*x*	*Error*	*Bed*	*Good*
	Состояние	0	1	2	3	4	5	6
		S	A	B	C	D	E	F
Вход	***u***	1	4	2	2	4	-	-
	n	4	2	2	2	4	-	-
	d	4	4	3	3	4	-	-
	$\boldsymbol{\rho}$	5	5	5	6	5	-	-
	$\boldsymbol{\sigma}$	4	4	4	2	4	-	-

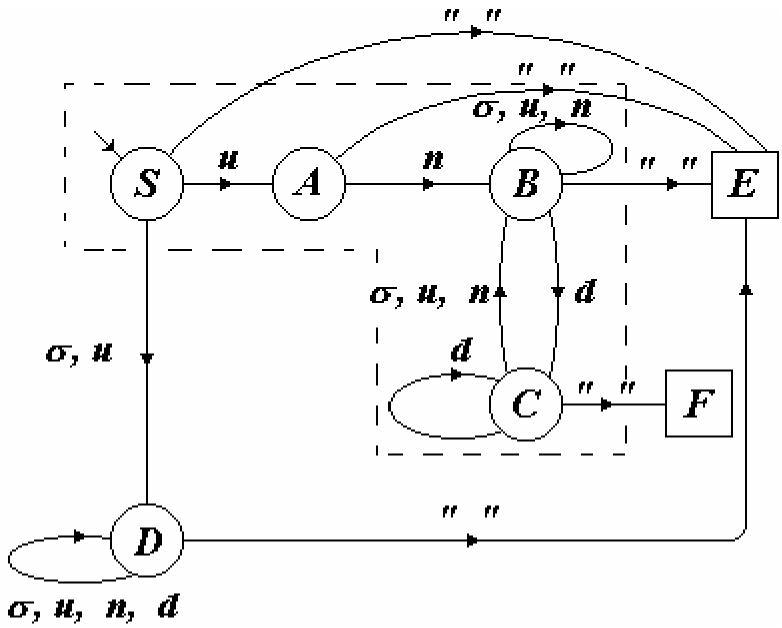

Рисунок 2.4 – Граф минимального конечного автомата

Вариант программы, реализующей конечный автомат, подсчитывающий числа «правильных» слов в произвольном текстовом файле, на языке программирования **С**и выглядит следующим образом.

```
main()
{
char c;                /* текущая литера */
int i, n;
file * text;
int mp[5][6] =    {{1, 4, 2, 2, 4}, // управляющая
                                    // таблица
                   {4, 2, 2, 2, 4},
                   {4, 4, 3, 3, 4},
                   {5, 5, 5, 6, 5},
                   {4, 4, 2, 2, 4}};
text = fopen('xx.txt', 'r'); // исходный текст в файле
```

```
i=0; n=0;         /* i - номер состояния КА, n - число
правильных слов */
while ((c = getc(text))! = EOF)
     {
     switch(c)
          {
          case 'u': i = mp[0][i]; break;
          case 'n': i = mp[1][i]; break;
          case 'd': i = mp[2][i]; break;
          case '  ': i = mp[3][i]; break;
          default:  i = mp[4][i];break;
          }
     if (i ==6) {i = 0; n++;}
     if (i ==5) {i = 0;}
     }
printf("количество символов =%d", n);
fclose(text);
}
```

Язык, анализируемый конечным автоматом, может быть описан грамматиками класса 3 в праволинейной или леволинейной формах [9], которые могут быть построены согласно формальной системе правил соответствия конечного автомата и формальной грамматики. Для праволинейной грамматики[6]: а) каждый нетерминал грамматики соответствует состоянию КА; б) каждому правилу вида $\mathbf{W}\rightarrow v$ соответствует дуга, направленная из начального к состоянию **W**, помеченная символом v; в) каждому правилу вида $\mathbf{W}\rightarrow\mathbf{R}v$ соответствует дуга из состояния **R** к состоянию **W**, помеченная символом v; где $\mathbf{W},\mathbf{R}\in V_H$, $v\in V_T$. Для построения леволинейной грамматики меняются позиции символов в правилах грамматики и направление «входящий – исходящий» для дуг.

Грамматики, язык которых есть цепочки латинских литер с правильными головой ***un*** и хвостом ***d***, построены для выделенной пунктиром части рисунка 2.4 – графа конечного автомата.

Таблица 2.3 – Автоматные грамматики

Праволинейная грамматика	Леволинейная грамматика
Аксиома грамматики: **C**	Аксиома грамматики: **S**
Тип разбора, осуществляемого конечным автоматом: *восходящий*.	Тип разбора, осуществляемого конечным автоматом: *нисходящий*.
$\mathrm{C}\rightarrow\mathrm{C}\boldsymbol{d}\mid\mathrm{B}\boldsymbol{d}$	$\mathbf{S}\rightarrow u\mathbf{A}$
$\mathrm{B}\rightarrow\mathrm{A}\boldsymbol{n}\mid\mathrm{C}\boldsymbol{u}\mid\mathrm{C}\boldsymbol{\sigma}\mid\mathrm{C}\boldsymbol{n}$	$\mathbf{A}\rightarrow n\mathbf{B}$
$\mathrm{A}\rightarrow\boldsymbol{u}$.	$\mathbf{B}\rightarrow d\mathbf{C}\mid d$
	$\mathbf{C}\rightarrow d\mathbf{C}\mid u\mathbf{B}\mid n\mathbf{B}\mid\sigma\mathbf{B}$.

Упражнения

2.1. Построить КА, восстановить грамматику, составить регулярные выражения, соответствующие цепочкам:

2.1.1. VINA, VINO, VINOGRAD, VINOGRADAR, VINODEL;

2.1.2. REAL, READ, REGISTER, RECORD;

2.1.3. ЛАССО, ЛАНЬ, ЛОСЬ, ЛОСОСЬ [8, с. 52];

2.1.4. оператор GOTO и GO TO, между слогами которого может быть произвольное число пробелов [8, с. 65];

2.1.5. построенным произвольным образом из подцепочек GO, GOTO, TOO, ON. Возможны повторения цепочек, но не их пересечения (например, GOON, GOTOON, GOTOOON) [8, с. 69];

2.1.6. изображающим химические формулы, составленные над алфавитом $\sum$={H, C, CL, N, O, S, SI, SN, цифры 0…9}. Формулы не обязательно соответствуют реальным веществам, элементы и радикалы отделяются запятыми, например $H_2O \Rightarrow$H2,O и $H_2SO_4 \Rightarrow$H2,S,O4. [8, с. 71];

2.1.7. над алфавитом $\{0,1\}$, в которых за каждой единицей непосредственно следует нуль [6, с.84];

2.1.8. над алфавитом $\{0,1\}$, в которых число единиц чётное, а нулей нечётное [8, с. 65];

2.1.9. над алфавитом $\{0,1\}$, в которых между единицами чётное число нулей [8, с. 65];

2.1.10. над алфавитом $\{0,1\}$, в которых число вхождений пары 00 нечётное, допускается вложение пар друг в друга [8, с. 65];

2.1.11. над алфавитом $\{0,1\}$, в которых за каждым вхождением пары 11 следует 0 [8, с. 65];

2.1.12. над алфавитом $\{0,1\}$, в которых каждый третий символ единица [8, с. 65];

2.1.13. над алфавитом $\{0,1\}$, в которых имеется хотя бы одна единица [8, с. 65];

2.1.14. над алфавитом $\{0,1\}$, в которых допускаются две цепочки 01 и 01000[8, с. 65];

2.1.15. над алфавитом $\{0,1\}$, которые начинаются с нуля и оканчивающаяся единицей [8, с. 65];

2.1.16. над алфавитом $\{0,1\}$, в которых не содержатся ни одной единицы [8, с. 65];

2.1.17. над алфавитом $\{0,1\}$, которые содержат всего три единицы [8, с. 65];

2.1.18. над алфавитом $\{0,1\}$, в которых перед и после каждой единицы стоят нули[8, с. 65].

2.2. Построить МДКА по регулярному выражению, определить грамматику и цепочки языка

2.2.1. $(\{ab\}\vee b\{a\})a$;
2.2.2. $\{ab\vee\{b\}\}ba\vee b$;
2.2.3. $\{\{\{b\}a\}ba\}$;
2.2.4. $a\,(ba\vee b)\vee b\{a\}$;
2.2.5. $\{ab\vee a\}a\{b\}a$;
2.2.6. $\{1\}\{01\{1\}\}$;
2.2.7. $\{10\}(1\vee 0)110\{0\}$;
2.2.8. $\{\{0\vee 1\}\vee(11)\}$, [6, с. 84];
2.2.9. $((a\vee b)\vee\{a\vee c\})\vee def$, [6, с. 84];
2.2.10. $((+\vee -)d\vee d)\{d\}.d\{d\}(E(+\vee -)\vee E)(d\vee dd\vee ddd)$;
2.2.11. $a\{ba\vee b\}\vee b$;
2.2.12. $\{ab\vee baa\}ba\vee b$;
2.2.13. $\{\{\{ba\}a\}ab\}$;
2.2.14. $\{101\}\{010\}$, [11, с. 52];
2.2.15. $\{101\}\{110\}$, [11, с. 52].

2.3. Построить КА, распознающий цепочки, порождаемые право- и леволинейными грамматиками

2.3.1.	$\mathbf{S}\to\mathbf{A}a\vert\mathbf{B}b$ $\mathbf{A}\to\mathbf{A}a\vert a$ $\mathbf{B}\to\mathbf{B}b\vert b.$	2.3.2.	$\langle\mathbf{Z}\rangle\to\langle\mathbf{cod}\rangle b$ $\langle\mathbf{cod}\rangle\to\langle\mathbf{cod}\rangle 1\vert\langle\mathbf{cod}\rangle 0$ $\langle\mathbf{cod}\rangle\to 1\vert 0.$
2.3.3. [6, с. 84]	$\mathbf{Z}\to\mathbf{A}0$ $\mathbf{A}\to\mathbf{A}0\vert\mathbf{Z}1\vert 0.$	2.3.4. [9, с. 61]	$\mathbf{A}\to a\mathbf{B}\vert a\mathbf{D}\vert a$ $\mathbf{B}\to a\mathbf{B}\vert b\vert b\mathbf{D}$ $\mathbf{D}\to d\mathbf{D}.$
2.3.5. [11, с. 206]	$\mathbf{S}\to x\mathbf{V}\vert y\mathbf{T}$ $\mathbf{V}\to x\mathbf{V}\vert b\mathbf{B}\vert b$ $\mathbf{T}\to y\mathbf{T}\vert b\mathbf{B}\vert b$ $\mathbf{B}\to b\mathbf{B}\vert b.$	2.3.6. [11, с. 206]	$\mathbf{A}\to 1\mathbf{A}\vert 1\mathbf{B}\vert 1$ $\mathbf{B}\to 0\mathbf{B}\vert 1\mathbf{B}\vert 1.$
2.3.7. [11, с. 210]	$\mathbf{A}\to 0\mathbf{A}\vert 1\mathbf{A}\vert 1\mathbf{B}$ $\mathbf{B}\to 0\mathbf{C}\vert 1\mathbf{C}$ $\mathbf{C}\to 0\mathbf{C}\vert 1\mathbf{C}\vert 1\vert 0.$	2.3.8.	Пример 1.16.7..

2.4. По графам конечных автоматов, представленным на рисунке 2.5, построить соответствующие грамматики и регулярные выражения, описывающее языки.

2.5. Последовательность считается правильной, если внутри ее содержатся (в любом сочетании) цепочки 335 и 028. Построить МДКА, проверяющий корректность цепочки.

2.6. Построить, если возможно, МДКА, поверяющий принадлежность цепочек языкам, приведенным в условиях задачи 1.9.

2.7. Товарный поезд состоит из одного или двух локомотивов ***Л***, одной или более платформ ***П*** и следующего за ними вагона с охраной ***О***. Построить конечный автомат, распознающий товарный поезд [11, с. 59].

2.8. Построить конечный автомат, контролирующий чётность единиц в строке над алфавитом {0,1} [11, с. 42].

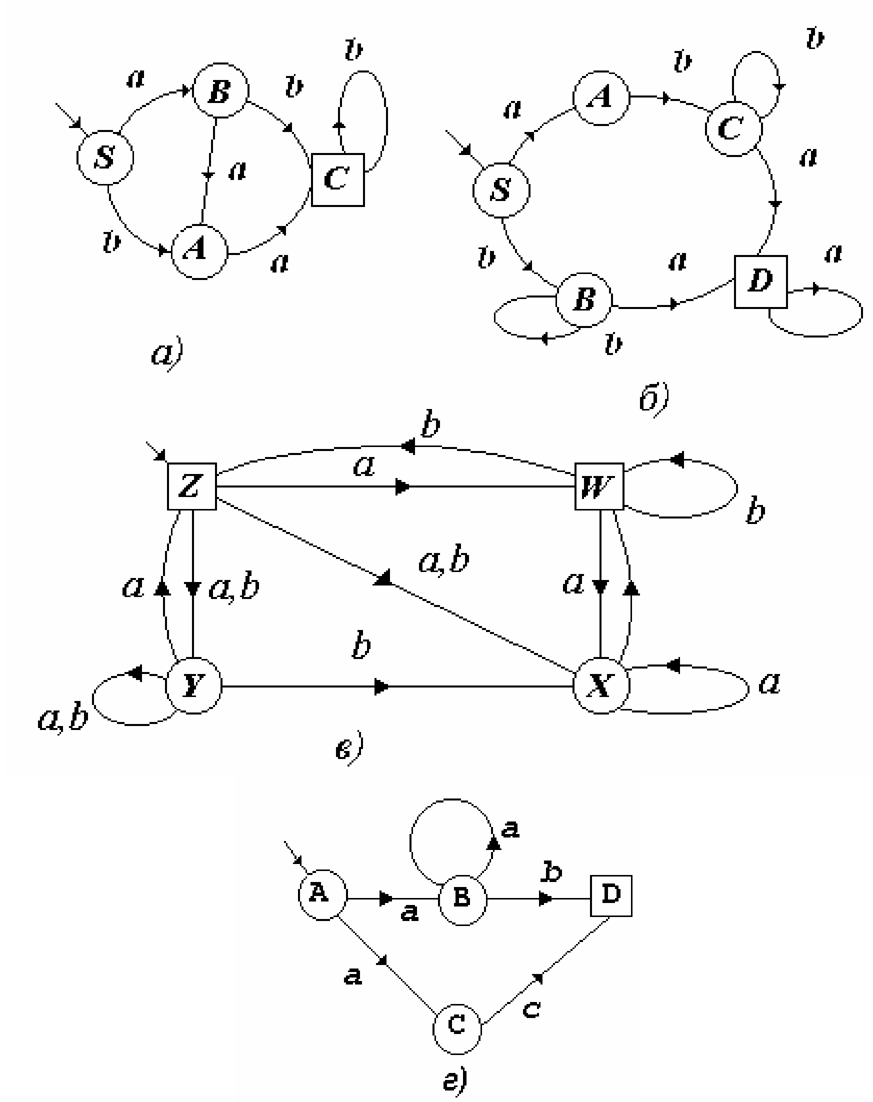

Рисунок 2.5 – Граф конечного автомата

2.9. Для конечного автомата, заданного совмещённой таблицей переходов и сигналов, найти: а) самую короткую цепочку, допускаемую автоматом; б) найти любые четыре цепочки, допускаемые автоматом; в) найти любые четыре цепочки, отвергаемые автоматом [8, с. 65].

Сигнал		не доп.	не доп.	не доп.	не доп.	доп.	доп.
Состояние		A	B	C	D	E	F
Вход	0	D	A	A	B	B	E
	1	A	C	F	C	C	A

2.10. Найти множество цепочек, распознаваемых конечным автоматом [8, с. 66];

2.10.1.

Сигнал		не доп.	не доп.	доп.
Состояние		A	B	C
Вход	0	B	B	C
	1	A	C	C

2.10.2.

Сигнал		не доп.	Доп.	не доп.
Состояние		A	B	C
Вход	0	B	C	C
	1	C	B	C

2.10.3.

Сигнал		не доп.	доп.	доп.	не доп.
Состояние		A	B	C	D
Вход	0	B	D	C	D
	1	C	B	D	D

2.10.4.

Сигнал		не доп.	не доп.	доп.	не доп.
Состояние		A	B	C	D
Вход	0	B	D	C	D
	1	A	C	D	D

2.10.5.

Сигнал		не доп.	не доп.	не доп.	не доп.	доп.	не доп.
Состояние		A	B	C	D	E	F
Вход	0	F	F	C	C	F	F
	*	F	C	D	C	F	F
	1	B	F	C	E	F	F

2.11. Найти множество цепочек минимальной длины, распознаваемых конечным автоматом, таких, что под их действием а) каждое состояние имеет место хотя бы раз; б) каждый переход выполняется хотя бы раз [8, с. 66];

2.11.1.

Сигнал		не доп.	доп.	не доп.	не доп.	доп.	доп.
Состояние		A	B	C	D	E	F
Вход	0	C	D	A	A	F	E
	1	B	B	E	A	E	A

2.11.2.

Сигнал		не доп.	не доп.	доп.	доп.	доп.	не доп.
Состояние		A	B	C	D	E	F
Вход	0	A	B	C	D	B	F
	1	E	D	F	B	F	C

2.12. Для пары конечных автоматов A и B, заданных таблицей переходов, найти хотя бы одну различающую цепочку, если она существует [8, с.67]

		Конечный автомат A					Конечный автомат B			
Сигнал		доп.	не доп.	доп.	не доп.		доп.	не доп.	доп.	не доп.
Состояние		A	B	C	D		A	B	C	D
Вход	0	A	C	B	A		A	A	B	C
	1	B	D	A	B		D	D	A	B

2.13. Найти недостижимые состояния конечного автомата [8, с. 67]

Сигн.		не доп.	доп.	не доп.	доп.	не доп.	Доп.	не доп.	доп.	не доп.	доп.
Сост.		A	B	C	D	E	F	G	H	I	J
Вход	0	C	J	J	F	E	D	H	G	D	B
	1	E	E	A	A	J	I	A	J	F	H
	2	G	G	H	G	H	A	J	B	G	G

2.14. Опишите множество цепочек, распознаваемых недетерминированным конечным автоматом [8, с. 69]

2.14.1.

Сигнал		не доп.	не доп.	доп.
Состояние		A	B	C
Вход	0	A,B	-	C
	1	-	B,C	-

2.14.2.

Сигнал		не доп.	не доп.	доп.
Состояние		A	B	C
Вход	0	B	-	-
	1	-	C,A	-

2.15 Построить детерминированный конечный автомат по недетерминированному конечному автомату [8, с. 69]

2.15.1.

Сигнал		не доп.	не доп.	не доп.	доп.
Состояние		A	B	C	D
Вход	*a*	B	C,D	B	-
	c	-	B	-	-

2.15.2.

Сигнал		не доп.	не доп.	доп.
Состояние		1	2	3
Вход	*a*	2,3	-	1
	b	2	1	3

2.16. Для заданных ниже конечных автоматов A_1 и A_2 построить детерминированный конечный автомат, воспринимающий цепочки вида: а) $S_1 \cup S_2$, б) $S_1 \cap S_2$, , где S_i, $i=1,2$ - множество цепочек, допускаемых i –ым конечным автоматом [8, с. 70].

		Конечный автомат A_1				Конечный автомат A_2			
Сигнал		доп.	доп.	не доп.		Не доп.	не доп.	доп.	не доп.
Состояние		A	B	C		A	B	C	D
Вход	0	C	B	C		D	C	D	D
	1	B	C	C		B	C	D	D

4. *LL*(*k*)-ГРАММАТИКИ И РЕКУРСИВНЫЙ СПУСК

Цели и задачи раздела:

а) освоить методику проявления исследований формальных грамматик на свойства *LL*(*k*);

б) научиться выполнять преобразование формальной грамматики к *LL(k)* - виду;

в) осуществлять проектирование нисходящего распознавателя, основанного на *LL*(*k*) - свойствах грамматики.

Система ссылок на теоретические источники

Теоретические положения, подлежащие усвоению	Исчерпывающее изложение	Доходчивое изложение	Изложение, адаптированное к восприятию
1. Исследование формальных грамматик на свойство *LL*(*k*)	[3, 5, 9]	[5, 10]	[2, с. 24 - 25]
2. Эквивалентные преобразования грамматик	[10]	[10]	[2, с. 37 - 41]
3. Построение *LL*(*k*) распознавателя и его программирование	[3, 5, 10]	[5, 10]	[2, с. 25 - 26]

Примеры выполнения заданий

Осуществить проектирование нисходящего распознавателя, основанного на *LL*(*k*) ***- свойствах грамматики***.

Рекурсивный спуск применяется для специального класса грамматик, обозначаемого аббревиатурой *LL*(*k*) (от латинской записи *Left-Left*). *LL*(*k*)-грамматика обладает тем свойством, что любые *k* первых символов анализируемой строки однозначно определяют правило грамматики, использованное для их вывода из аксиомы. Алгоритм рекурсивного спуска заключается в последовательности применения процедур, распознающих терминальные и нетерминальные символы грамматики, в том порядке, в котором они размещены в правых частях соответствующих продукций.

Рассмотрим процедуру определения грамматики на принадлежность к классу *LL*(*k*) – грамматик на примере.

$\mathbf{S}\rightarrow a\mathbf{AB}|b\mathbf{S}$

$\mathbf{A}\rightarrow a\mathbf{A}|b\mathbf{S}$

$\mathbf{B}\rightarrow \mathbf{AB}|c$.

Для начала, грамматика исследуется для k=1, результаты исследования отображены в таблице 4.1. Согласно алгоритму, необходимо для каждой продукции грамматики найти головы цепочек из k (в данном случае, из одного) символов, которые могут быть получены в процессе применения данного правила (строки 1 – 4, 6). Если первый символ правой части продукции является нетерминальным, то ей ставится в соответствие объединённое множество, сконструированное для продукций, имеющих в левой части означенный нетерминал (строка 5).

Таблица 4.1 - Исследование грамматики на $LL(1)$ - свойства

Продукции	Цепочки
1. $\mathbf{S}\to a\mathbf{AB}$	$\{a\}$
2. $\mathbf{S}\to b\mathbf{S}$	$\{b\}$
3. $\mathbf{A}\to a\mathbf{A}\vert$	$\{a\}$
4. $\mathbf{A}\to b\mathbf{S}$	$\{b\}$
5. $\mathbf{B}\to\mathbf{AB}$	$\{a, b\}$
6. $\mathbf{B}\to c$	$\{c\}$

Если множества, соответствующие продукциям с одинаковыми правыми частями, не пересекаются, то это $LL(k)$ грамматика. Данный факт просматривается в таблице. Понятно, что если цепочки отличаются по нескольким первым символам (головам), то остальные символы в цепочках (хвосты) можно не рассматривать, и $LL(1)$ грамматика будет $LL(2)$ грамматикой и т.д.

Напишем нисходящий распознаватель, основанный на *LL(1)* свойствах. Его структура будет соответствовать структуре продукций грамматики. Пусть, имена процедур анализатора будут начинаться символом *P* и иметь вид *Px*, где *x*- символ алфавита, распознаваемый (определяемый) процедурой. Возможный вид программы следующий.

```
int pA(void), pB(void), pS(void), pa(void), pb(void),
pc(void);
char symbol;
void main(void)
{
    scanf("%S", &symbol);
    if(PS()==0) printf("\n Ошибка ");
    else printf("\n Ok ");
}
int PS(void)
{
    switch (symbol)
    {
```

```
        case 'a': return pa()*PA()*PB();
        case 'b': return pb()*PS();
        default:   return 0;
    }
}
int PA(void)
{
    switch (symbol)
    {
        case 'a': return pa()*PA();
        case 'b': return pb()*PS();
        default:   return 0;
    }
}
int PB(void)
{
    switch (symbol)
    {
        case 'a': return PA()*PB();
        case 'b': return PA()*PB();
        case 'c': return pc();
        default: return 0;
    }
}
int pa(void)
{
    if (symbol!= 'a') then return 0;
    else { scanf("%S", &symbol); return 1; }
}
int pb(void)
{
    if (symbol!= 'b') return 0;
    else { scanf("%S", &symbol); return 1; }
}
int pc(void)
{
    if (symbol!= 'c') return 0;
    else { scanf("%S", &symbol); return 1; }
}
```

Упражнения

4.1. Показать, что грамматика является *LL(1)* грамматикой и построить рекурсивный распознаватель ее языка.

$\mathbf{S} \to o\mathbf{AB} | \textit{ё}\mathbf{S}$

$\mathbf{A} \to o\mathbf{A} | \textit{ё}\mathbf{S}$

$\mathbf{B} \to \mathbf{AB} | \textit{й}.$

4.2. Какие из ниже перечисленных грамматик являются грамматиками типа *LL(1)?*

4.2.1.
$\mathbf{S} \to \mathbf{Z} | \mathbf{U}$
$\mathbf{Z} \to 0\mathbf{U} | \mathbf{U}$
$\mathbf{U} \to 1\mathbf{U} | 1.$

4.2.2.
$\mathbf{S} \to x\mathbf{Z}x\mathbf{U} | y\mathbf{Z}y\mathbf{U}$
$\mathbf{Z} \to \mathbf{S} | x\mathbf{U}$
$\mathbf{U} \to x\mathbf{U} | x.$

4.2.3.
$\mathbf{S} \to \mathbf{UW}$
$\mathbf{U} \to \mathbf{W}o | \mathbf{Z}$
$\mathbf{W} \to \mathbf{X}a | \mathbf{X}$
$\mathbf{X} \to \textit{й} | \mathbf{Z}$
$\mathbf{Z} \to o | a.$

4.2.4. [10, с. 104]
$\mathbf{S} \to \mathbf{AB} | b\mathbf{S}$
$\mathbf{A} \to a\mathbf{A} | b\mathbf{B}.$
$\mathbf{B} \to \mathbf{AB} | c.$

4.2.5. [10, с. 109]
$\mathbf{S} \to \mathbf{A} | \mathbf{B}$
$\mathbf{A} \to a\mathbf{A} | a$
$\mathbf{B} \to b\mathbf{B} | b.$

4.2.6. [10, с. 109]
$\mathbf{S} \to \mathbf{AB}$
$\mathbf{A} \to \mathbf{B}a | \varepsilon$
$\mathbf{B} \to \mathbf{C}b | \mathbf{C}$
$\mathbf{C} \to c | \varepsilon.$

4.2.7. [9, с. 109]
$\mathbf{S} \to a\mathbf{A}a\mathbf{B} | b\mathbf{A}b\mathbf{B}$
$\mathbf{A} \to \mathbf{S} | cb$
$\mathbf{B} \to c\mathbf{B} | a.$

4.2.8. [11, с. 85]
$\mathbf{S} \to \mathbf{AB} | \mathbf{PQ}x$
$\mathbf{A} \to xy | m$
$\mathbf{B} \to b\mathbf{C}$
$\mathbf{C} \to b\mathbf{C} | \varepsilon$
$\mathbf{P} \to p\mathbf{P} | \varepsilon$
$\mathbf{Q} \to q\mathbf{Q} | \varepsilon.$

4.2.9. [3, с. 380]
$\mathbf{S} \to a\mathbf{AS} | b$
$\mathbf{A} \to b\mathbf{SA} | a.$

4.2.10. Пример 1.11.5..

4.3. Покажите, что следующая грамматика является LL(2) грамматикой и постройте соответствующий ей лексический анализатор

4.3.1.
$\mathbf{Z} \to a\mathbf{UZ} | \mathbf{U}q\mathbf{Z}t$
$\mathbf{U} \to tq\mathbf{U} | a.$

4.3.2. [10, с.109]
$\mathbf{S} \to a\mathbf{AS} | \mathbf{A}b\mathbf{S}c | \varepsilon$
$\mathbf{A} \to cb\mathbf{A} | a.$

4.3.3. [10, с.109]
$\mathbf{S} \to a\mathbf{A}aa | b\mathbf{A}ba$
$\mathbf{A} \to b | \varepsilon.$

4.4. Покажите, что нижеприведённая грамматика является LL(3) грамматикой и постройте соответствующий ей лексический анализатор.

4.4.1. [9, с.110]
$\mathbf{S} \to a\mathbf{BA} | b\mathbf{B}b\mathbf{A}$
$\mathbf{A} \to ab\mathbf{A} | c$

4.4.2. [3, с. 400]
$\mathbf{S} \to a\mathbf{A}a\mathbf{B} | b\mathbf{A}b\mathbf{B}$
$\mathbf{A} \to a | ab$

$\mathbf{B} \to a|ab.$ $\mathbf{B} \to aba.$

4.5. Преобразовать грамматику к виду LL(1).

$\mathbf{E} \to \mathbf{E}+\mathbf{T}|\mathbf{T}$

$\mathbf{T} \to \mathbf{T}*\mathbf{F}|\mathbf{F}$

$\mathbf{F} \to (\mathbf{E})|<data>|<iden>.$

4.6. Построить LL(1) – грамматику, язык которой [11, с. 85]:

4.6.1. $L = \{0^n a 1^{2n}, n \geq 0\}$;

4.6.2. определён на множестве *{0,1}* и не содержит две последовательные единицы;

4.6.3. определён на множестве *{0,1}* и содержит равное число нулей и единиц.

5. ПОСТРОЕНИЕ ОТНОШЕНИЙ ПРЕДШЕСТВОВАНИЯ

Цели и задачи раздела:

а) научиться исследовать грамматики на свойство предшествования;

б) освоить технику построения отношений простого предшествования;

в) освоить процедуру построения отношений операторного предшествования;

г) выполнять эквивалентные преобразования грамматики с целью придания ей необходимых свойств;

д) по заданным отношениям предшествования выполнять построение функций предшествования.

Система ссылок на теоретические источники

Теоретические положения, подлежащие усвоению		Исчерпывающее изложение	Доходчивое изложение	Изложение, адаптированное к восприятию
1.	Определение свойства предшествования	[3, 6, 10]	[5, 6, 9]	[2, с. 27]
2.	Алгоритм построения простого предшествования	[3, 6, 10]	[5, 6, 9]	[2, с. 28 - 29, с, 30 - 33]
3.	Алгоритм построения операторного предшествования	[3, 6, 10]	[5, 6]	[2, с. 5]
4.	Алгоритм построения функций предшествования	[3, 6, 9]	[6]	
5.	Эквивалентные преобразования грамматик	[9]	[9]	[2, с. 29 - 30, с. 34 - 36]

Примеры выполнения заданий

Построить отношения простого предшествования и, если возможно, функции предшествования для грамматики

S→*(***R**|*a*

R→*Sa).*

Существует несколько описаний алгоритма [3 – 6, 8, 9, 11] для построения отношений предшествования. Воспользуемся изложенным в [6]. Согласно алгоритму, нам необходимо построить серию булевых матриц размерностью $n \times n$, где n –мощность словаря грамматики $V. = (V_N \cup V_T)$ Индексами строк выступают “левые” символы R, а индексами столбцов – “правые” S.

1. Построим матрицу EQ(≡). Для этого необходимо просмотреть все правые части продукций, и, для каждой пары символов, стоящих рядом, поставить единицу в соответствующей позиции матрицы.

	S	**R**	*a*	*(*	*)*
S			1		
R					
a					1
(		1			
)					

2. Строим вспомогательную матрицу F отношения FIRST, которое определяется следующим образом: $F(\mathbf{R}, S) = 1$ для всех правил вида $\mathbf{R} \rightarrow s\omega$, $\mathbf{R} \in V_N$, $s \in V$, $\omega \in V^*$.

	S	**R**	*a*	*(*	*)*
S			1	1	
R	1				
a					
(					
)					

3. Строим вспомогательную матрицу L отношения $LAST$, которое определяется по правилу: $L(\mathbf{R}, \mathrm{S})=1$, $\mathbf{R} \rightarrow \omega s$, $\mathbf{R} \in V_N$, $s \in V$, $\omega \in V^*$.

	S	**R**	*a*	*(*	*)*
S		1	1		
R					1
a					
(					
)					

4. Проведём построение транзитивного замыкания отношений F^+ и L^+. Указанные отношения вычисляются по рекуррентному алгоритму

$$A^+ = \mathrm{A}^1 \cup A^2 \cup A^3 \cup \ldots;$$
$$A^1 = A,\ A^2 = A \times A,\ A^{\mathrm{n}} = A^{n-1} \times A$$

вручную, либо, используя ЭВМ по алгоритму Воршалла (Warshall S.) [6, с. 56 - 57]:

```
for (i=0; i<n-1; i++)
     for (j=0; j<n-1; j++)
          if (A[j][i] = = 1)
               for (k=0; k<n-1; k++)
                    A[j][kj] Θ = A[i][k];
```

Символом Θ обозначен оператор булевого суммирования. Так как вопросы прикладного программирования Читателем вполне освоены, продемонстрируем нахождение транзитивного замыкания вручную.

$$F^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$L^2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$L^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Дальнейшее возведение в степень лишено смысла, поэтому

$$F^+ = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; L^+ = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

5 Вычисляем отношение предшествования $<\bullet$ как матричное выражение $EQ \times F^+$.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

6 Вычисляем отношение предшествования •> как матричное выражение $(L^+)^T \times EQ \times (I + F^+)$, где I – единичная матрица.

$$(L^+)^T \times EQ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$I + F^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Окончательно имеем

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

7 Компонуем матрицы отношений в единую матрицу и проверяем (убеждаемся) в единственности отношения для каждой пары символов. Видно, что грамматика является грамматикой (простого) предшествования. Попытаемся построить функции предшествования для данной матрицы отношений.

	S	**R**	*a*	*(*	*)*
S			≡		
R			•>		
a			•>		≡
(	<•	≡	<•	<•	
)			•>		

Функции строятся по следующему алгоритму [6, 138 - 139].

1. Конструируется блочная матрица вида

$$B = \begin{bmatrix} O & GE \\ LE^{T} & O \end{bmatrix},$$

в которой GE – объединённая матрица отношений •> и ≡, LE^{T} – транспонированная объединённая матрица отношений <• и ≡.

$$LE = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$GE = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

2. Находится транзитивно-рефлексивное замыкание $B^{*}=E+B^{+}$, где E – единичная матрица, B^{+} - транзитивное замыкание B.

$$B = \begin{bmatrix} 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0 \end{bmatrix}, B^2 = \begin{bmatrix} 1&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&1&0&1 \end{bmatrix},$$

$$B^3 = \begin{bmatrix} 0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&1&0&1\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&1&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0 \end{bmatrix}, B^4 = \begin{bmatrix} 1&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&1&0&1 \end{bmatrix}.$$

Начиная с пятой итерации, матрицы B^5, результаты вычислений будут чередоваться и равны, соответственно, B^3 и B^4. Поэтому, транзитивно-рефлексивное замыкание определиться матрицами B … B^4.

$$B^* = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

3. Подсчёт единиц в матрице B* позволяет определить значения функций предшествования *f(R$_i$)* и *g(S$_i$)*.

$$F_i = f(R_i) = \sum_{j=1}^{2n} B^*_{i,j}; \quad i = \overline{1,n};$$

$$G_i = g(S_i) = \sum_{i=1}^{2n} B^*_{n+i,j}; \; i = \overline{1,n}.$$

4. Выполняется проверка непротиворечивости построенных функций исходным отношениям. Если противоречий нет, то функция построена правильно, в противном случае, функций предшествования не существует.

$$B^* = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} \left.\begin{matrix} 4 \\ 5 \\ 6 \\ 2 \\ 5 \end{matrix}\right\} F_i \\ \left.\begin{matrix} 3 \\ 2 \\ 4 \\ 3 \\ 6 \end{matrix}\right\} G_i \end{matrix}$$

Нижеприведённые данные свидетельствуют о существовании для данной грамматики функций предшествования.

	F_i	**S**	**R**	*a*	*(*	*)*
G_i		3	2	4	3	6
S	4			≡		
R	5			•>		
a	6			•>		≡
(	2	<•	≡	<•	<•	
)	5			•>		

Примечание. Алгоритм построения отношений операторного предшествования несколько отличается от алгоритма построения отношений простого предшествования и содержит следующие шаги.

1. Необходимо определить, является ли грамматика операторной, т.е., по определению [3 - 6], свободной от правил, в правой части которых наличествуют *смежные (рядом стоящие) нетерминалы*.

2. Осуществляется построение матриц EQ, L, F размером $n \times n$ по правилам, изложенным выше, и вычисляются транзитивные замыкания L^+ и F^+.

3. Выполняется построение матрицы отношения FT (First Terminal) размерности $n{\times}n$ по правилу FT(**R**, s) = 1, если существуют правила $\mathbf{R} \to s\beta$ или $\mathbf{R} \to \mathbf{Q}s\beta$, в которых $\mathbf{R}, \mathbf{Q} \in V_N$, $s \in \mathrm{V_T}$, $\beta \in V^*$.

4. Выполняется построение матрицы отношения LT (Last Terminal) размерности $n{\times}n$ по правилу LT(**R**, s)=1, если существуют правила $\mathbf{R} \to \alpha s$ или $\mathbf{R} \to \alpha\mathbf{Q}s$, в которых $\mathbf{R}, \mathbf{Q} \in V_N$, $s \in V_T$, $\alpha \in \mathrm{V}^*$.

5. Матрица отношения EQT (≡°) определяется визуально для пары терминальных символов EQT(r, s) при наличии правила $\mathbf{U} \to \alpha rs\beta$ или $\mathbf{U} \to \alpha r\mathbf{Q}s\beta$, где $\mathbf{U}, \mathbf{Q} \in \mathrm{V_H}$, $r,s \in V_T$, $\alpha,\beta \in V^*$, и имеет измерения, определяемые мощностью словаря терминалов V_T.

6. Вычисляются отношения операторного предшествования по формулам:

$$<^\circ = EQ \times (I + F^+) \times FT \text{ и}$$

$$^\circ\!\!>=[(I + L^+) \times LT]^{\mathrm{T}} \times \mathrm{EQ}.$$

7. Из матриц <°, °> и ≡° компонуется матрица той же размерности, что и EQT. Тут же выполняется проверка единственности отношений.

Данные операции не содержат принципиальных моментов, поэтому пример построения отношений операторного предшествования не рассматривается.

Упражнения

5.1. По деревьям грамматик определить отношения предшествования.

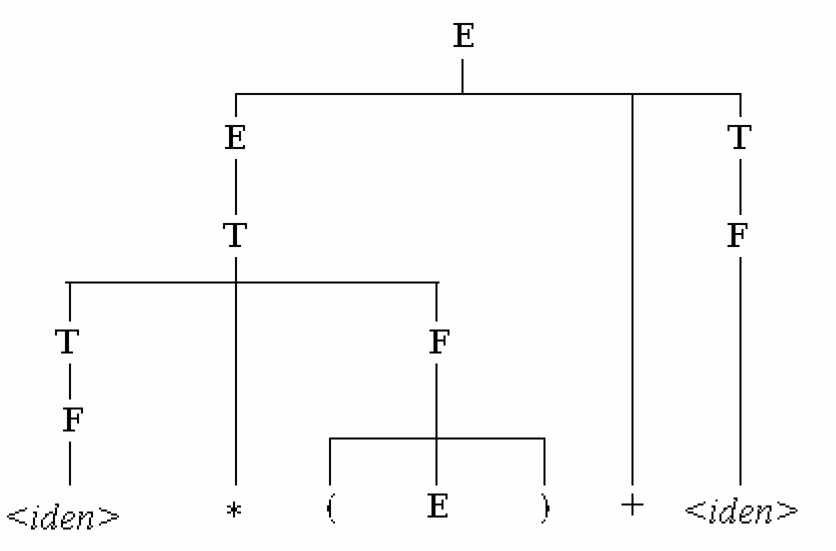

Рисунок 5.1 – Дерево представления грамматики

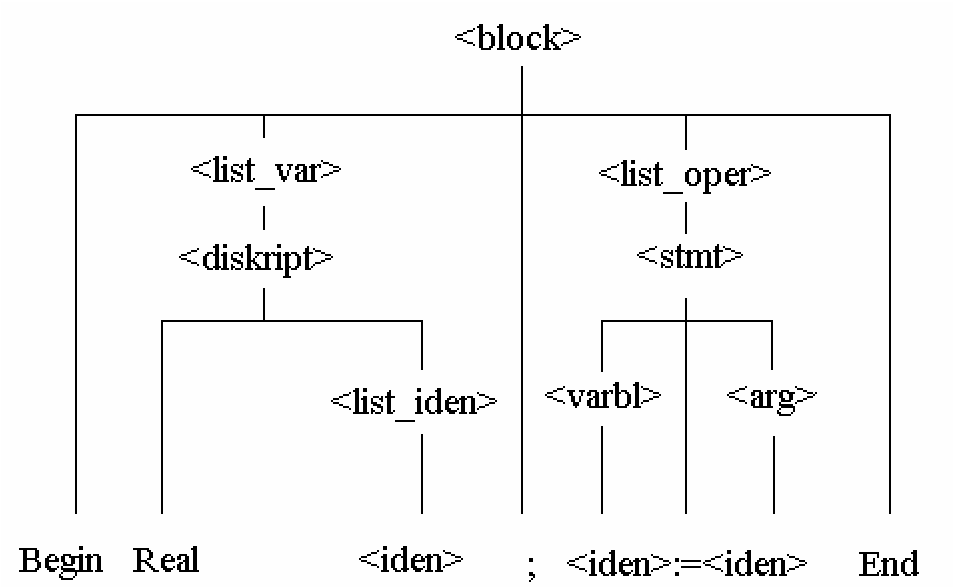

Рисунок 5.2 – Дерево представления грамматики

5.2. Построить, где это возможно, отношения простого и операторного предшествования, функции предшествования для грамматики.

5.2.1.	$\mathbf{S} \rightarrow (\mathbf{R} \mid a$	5.2.2.	$\mathbf{Z} \rightarrow b\mathbf{E}b$
	$\mathbf{R} \rightarrow \mathbf{S}a)$	[6, с.134]	$\mathbf{E} \rightarrow \mathbf{E}+\mathbf{T} \mid \mathbf{T}$

5.2.3. [6, с.134]

$\mathbf{Z} \to b\mathbf{Q}b$
$\mathbf{Q} \to \mathbf{E}|\mathbf{E}+\mathbf{T}|\mathbf{T}|<iden>$
$\mathbf{E} \to \mathbf{T}*\mathbf{T}$
$\mathbf{T} \to <iden>|(\mathbf{Q})$

5.2.4. [5, с. 131]

$\mathbf{S} \to \mathbf{S}+\mathbf{B}|\mathbf{B}$
$\mathbf{B} \to \mathbf{B}\times\mathbf{G}|\mathbf{B}$
$\mathbf{G} \to a|(\mathbf{S})$

5.2.5. [5, с. 131]

$\mathbf{S} \to \mathbf{A}$
$\mathbf{A} \to \mathbf{A}+\mathbf{B}|\mathbf{B}$
$\mathbf{B} \to \mathbf{B}*\mathbf{G}|\mathbf{G}$
$\mathbf{G} \to (\mathbf{A})|a$

5.2.6. [5, с. 137]

$\mathbf{S} \to \mathbf{A}|\mathbf{Ba}d$
$\mathbf{A} \to ab\mathbf{D}|c$
$\mathbf{D} \to ab|d$
$\mathbf{B} \to \mathbf{A}b$

5.2.7. [5, с. 149]

$\mathbf{A} \to \mathbf{A}aa|a$

5.2.8. [5, с. 149]

$\mathbf{S} \to a\mathbf{A}c|b$
$\mathbf{A} \to a\mathbf{S}c|b$

5.2.9. [5, с. 149]

$\mathbf{S} \to a\mathbf{A}d|b\mathbf{AB}$
$\mathbf{A} \to c\mathbf{A}|c$
$\mathbf{B} \to d$

5.2.10. [5, с. 149]

$\mathbf{S} \to a\mathbf{A}d|a\mathbf{B}c$
$\mathbf{A} \to b\mathbf{A}|b$
$\mathbf{B} \to \mathbf{B}b|b$

5.2.11. [5, с.153]

$\mathbf{A} \to a\mathbf{A}|ab$

5.2.12. [5, с.154]

$\mathbf{A} \to \mathbf{D}|\mathbf{B}$
$\mathbf{D} \to c\mathbf{D}|b$
$\mathbf{B} \to a\mathbf{B}|b$

5.2.13. [5, с.162]

$\mathbf{S} \to a\mathbf{A}d|b\mathbf{AB}$
$\mathbf{A} \to c\mathbf{A}|c$
$\mathbf{B} \to d$

5.2.14. [5, с.178]

$\mathbf{S} \to \mathbf{A}|\mathbf{D}$
$\mathbf{A} \to ab|a\mathbf{B}|\mathbf{A}b$
$\mathbf{D} \to \mathbf{DB}|\mathbf{A}$
$\mathbf{B} \to c$

5.2.15. [5, с.163]

$\mathbf{S} \to \mathbf{ABG}|\mathbf{A}a|a$
$\mathbf{A} \to \mathbf{A}a|a$
$\mathbf{G} \to a\mathbf{G}|a$
$\mathbf{B} \to a$

5.2.16. [5, с.163]

$\mathbf{S} \to \mathbf{AB}$
$\mathbf{A} \to \mathbf{A}a$
$\mathbf{B} \to a|b\mathbf{B}|b$

5.2.17. [5, с.163]

$\mathbf{S} \to b\mathbf{A}|a\mathbf{B}$
$\mathbf{A} \to c\mathbf{A}|c$
$\mathbf{B} \to c\mathbf{B}|c$

5.2.18. [6, с.49]

$\mathbf{S} \to \mathbf{V}$
$\mathbf{V} \to \mathbf{U}|\mathbf{V}+\mathbf{U}$
$\mathbf{U} \to \mathbf{W}|\mathbf{U}\text{-}\mathbf{W}|+\mathbf{W}$
$\mathbf{W} \to)\mathbf{V}*|($

5.2.19. [6, с.142]

$\mathbf{E} \to \mathbf{E}+\mathbf{T}|\mathbf{T}$
$\mathbf{T} \to \mathbf{T}*\mathbf{F}|\mathbf{F}$
$\mathbf{F} \to (\mathbf{E})|<iden>$

5.2.20. [6, с.142]

$\mathbf{E} \to \mathbf{D}$
$\mathbf{D} \to \mathbf{D}+\mathbf{U}|\mathbf{U}$
$\mathbf{U} \to \mathbf{T}$
$\mathbf{T} \to \mathbf{T}*\mathbf{F}|\mathbf{F}$
$\mathbf{F} \to (\mathbf{G})|<iden>$

5.2.21.

$\mathbf{S} \to \mathbf{A}\#$
$\mathbf{A} \to a\mathbf{ABC}|\mathbf{CB}$
$\mathbf{B} \to a\mathbf{B}|c$
$\mathbf{C} \to b$

5.2.22. [4, с. 9]

$\mathbf{S} \to a\mathbf{S}c|b\mathbf{S}c|c$

5.2.23. [4, с. 23]

E→E+A|A
A→T
T→T*F|F
F→(B|<*iden*>
B→E)

5.2.24. [4, с. 25]

S→SA|A
A→(S)|()

5.2.25. [4, с. 25]

S→E+T|+T|T
T→T*F|F
F→(E)|<*iden*>

5.2.26. [3, с. 457]

S→*a***SS***bc*

5.2.27. [3, с. 465]

S→*0***S***11|011*

5.2.28. [3, с. 473]

E→E|E+T
T→T*F|F
F→<*iden*>|*(***E***)*|<*iden*>*(***L***)*
L→L,E|E

5.2.29. [3, с. 474]

E→E|E+T
T→T*F|F
F→<*iden*>|*(***E***)*|
F→<*iden*>*(***L,E***)*|<*iden*>*(***E***)*
L→L,E|E

5.2.30. [3, с. 477]

S→*if* **E** *then* **S** *else* **S**
S→<*iden*>
E→E*!*<*data*>|<*iden*>

5.2.31. [3, с. 477]

S→AS|A
A→*(***S***)|()*

5.2.32. [3, с. 477]

S→SA|A
A→*(***S***)|()*

5.2.33. [3, с. 547]

S→*a***SA|***b***SA|***b*
A→*a*

5.2.34. [8, с. 127]

S→A#
A→*a***ABC|CB**
B→*a***B|C**
C→*b*

5.2.35. [4, с. 42]

S→S+I|I
I→*(***S***)|a(***S***)|a*

5.2.36. [4, с. 42]

S→*0***S***1|01*

5.2.37. [3, с. 42]

E→E+T|T
T→T*F|F
F→F^P|P
P→(E)|<*iden*>

5.2.38. [4, с. 84]

S→ABAC
A→*a***D**
B→*b|c*
D→D*0|0*

5.2.39. [4, с. 114]

S→L=R|R
L→*R|*a*
R→L

5.2.40. [12, с. 547]

P→(E)|<*iden*>→
E→T|E+T
T→P|T*P

5.3. Не меняя языка, преобразовать данную грамматику в грамматику предшествования

5.3.1. [6, с.144]

<**Описание**>::=<**Тип**><**Список**>
<**Тип**>::=*REAL|INTEGER|BOOLEAN*
<**Список**>::=<*iden*>|<**Список**>,<*iden*>

5.3.2. **S**::=<**Логич_выр**>
[6, с.144] <**Логич_выр**>::=<**Логич_iden**>:=<**Логич_выр**>|
<**Ар_выр**>=<**Ар_выр**>
<**Ар_выр**>::=<**Ар_iden**>:=<**Ар_выр**>|<**Ар_терм**>
<**Ар_терм**>::=<**Ар_терм**>**-**<**Ар_первичное**>|<**Ар_первичное**>
<**Ар_первичное**>::=<**Ар_iden**>|(<**Ар_выр**>)|@(<**Логич_выр**>)
<**Ар_iden**>::=<*iden*>
<**Логич_iden**>::=<*iden*>

5.3.3. Грамматику из примера 5.2.2.7

ЗАКЛЮЧЕНИЕ

Настоящее издание ни в коей мере не исчерпывает всех аспектов направления, связанного с теориями формальных грамматик, абстрактных автоматов и компиляции, в связи с построением на их основах лексических и синтаксических анализаторов, а, скорее, напоминает задачник, являющийся более подспорьем преподавателю, нежели пособием студенту. Тем не менее, неоспорим факт, что все преподаватели были студентами, а отдельные студенты непременно станут преподавателями. Поэтому авторы вправе рассчитывать если не на энтузиазм, то на понимание аудитории, что, в конце концов, обеспечит усваивание положений дисциплины на должном уровне, в объёмах, определённых паспортом специальности. Дидактические материалы могут быть полезны при проведении занятий по дисциплинам «Элементы теории конечных автоматов», «Системное программное обеспечение» и др. как кафедры Информационных систем, так и родственных кафедр.

ОТВЕТЫ К ОТДЕЛЬНЫМ УПРАЖНЕНИЯМ

Формальные грамматики и их элементы

1.1.1. $V_N=\{\mathbf{A}\}$, $V_T=\{a\}$, аксиома грамматики - нетерминальный символ **A**, язык представляется цепочками вида $a^{2n-1}, n=1,2,\ldots$. 1.1.2. $V_N=\{\mathbf{A}\}$, $V_T=\{a\}$, аксиома грамматики - нетерминальный символ **A**, язык представляется цепочками вида $a^{2n+1}, n=0,1,2,\ldots$ 1.2.1.0 Один из возможных вариантов представлен в упражнении 1.15.8.. 1.2.1.1 $R=\{\mathbf{S}\to\mathbf{AB};\ \mathbf{A}\to a\mathbf{A}|a;\ \mathbf{B}\to b\mathbf{B}|b\}$, $V_T=\{a,\ b\}$, $V_N=\{\mathbf{S},\ \mathbf{A},\ \mathbf{B}\}$. 1.3.1. Является. 1.3.2. Не является, обратите внимание на правило № 3. 1.4.1. Контекстно-свободная (бесконтекстная) грамматика. 1.4.2. Контекстно-зависимая. 1.4.3. Не является ни контекстно-свободной ни контекстно-зависимой. 1.4.6. Контекстно-зависимая. 1.9.6. $R=\{\mathbf{S}\to 0\mathbf{S}|1\mathbf{S}|\varepsilon\}$, $V_T=\{0,\ 1\}$, $V_N=\{\mathbf{S}\}$. 1.11.1. Грамматики являются эквивалентными, но они неоднозначны. 1.11.2. Грамматики однозначны. 1.11.4. Грамматики эквивалентны, первая грамматика неоднозначна, вторая свободна от этого недостатка. 1.11.5. Грамматики эквивалентны. 1.11.7. Грамматики эквивалентны, 1-я неоднозначна, 2-я однозначна. 1.13.2. Данный пример получил название "кочующее ***else***", грамматика неоднозначна, ибо сентенциальной форме *IF b THEN IF b THEN a ELSE a* соответствуют сразу два синтаксических дерева *IF b THEN* {*IF b THEN a*} *ELSE a* и *IF b THEN* {*IF b THEN a ELSE a*}. 1.13.4. Грамматика неоднозначна. 1.14 $L=\{a^n b^n c^n, n\ge 1\}$. Построить КС-грамматику не представляется возможным. 1.17.4. Одно из возможных решений: $P=\{\mathbf{S}\to\mathbf{AB};\ \mathbf{S}\to\mathbf{AC};\ \mathbf{B}\to\mathbf{SC};\ \mathbf{A}\to 0;\ \mathbf{C}\to c\}$.

Построение конечных автоматов (КА)

2.2.15. Соответствующая леволинейная грамматика [11, с. 209] имеет вид $P=\{\mathbf{S}\to 1\mathbf{T}|0\mathbf{U}|\varepsilon;\ \mathbf{T}\to 0\mathbf{V};\ \mathbf{V}\to 1\mathbf{S};\ \mathbf{U}\to 1\mathbf{W};\ \mathbf{W}\to 0\mathbf{X}|0;\ \mathbf{X}\to 0\mathbf{U}\}$. 2.3.5. A(L)={A, B; Σ=(0,1); δ(A,1)=A; δ(A,1)=B; δ(B,0)=B; δ(B,1)=B; A; B}. 2.4.6., г) Леволинейная грамматика имеет вид $\mathbf{A}\to a\mathbf{B}|a\mathbf{C};\ \mathbf{B}\to a\mathbf{B}|b;\ \mathbf{C}\to c$. 2.8. A(L)={A, B; Σ=(0,1); δ(A,0)=A; δ(A,1)=B; δ(B,0)=B; δ(B,1)=A; A; B}.

LL(k)-грамматики и рекурсивный спуск

4.2.9. является *LL(k)*-грамматикой. 4.5 Указанная грамматика устранением левой рекурсии приводится к типу *LL(1)*. Система правил имеет вид [3, с. 386]: $P=\{\mathbf{E}\to\mathbf{TE_1};\ \mathbf{E_1}\to+\mathbf{TE_1}|\varepsilon;\ \mathbf{T}\to\mathbf{FT_1};\ \mathbf{T_1}\to*\mathbf{FT_1}|\varepsilon;\ \mathbf{F}\to(\mathbf{E})|\langle iden\rangle|\langle data\rangle\}$. Аналогичное решение даётся в [11, с. 213]. 4.6 Возможные решения выглядят так [11, с. 212]: 4.6.1. $\mathbf{S}\to 0\mathbf{S}11|a$. 4.6.2. Это условие примера 1.5.20; 4.6.3. Это условие примера 1.15.21.

Построение отношений предшествования

5.2.1. [2, с. 44]. 5.2.2. Не является грамматикой предшествования. 5.2.3. Не является грамматикой предшествования. 5.2.3. Не является грамматикой простого предшествования, является грамматикой операторного предшествования

	+	*	(	)	*iden*
+	°>	<°	<°	°>	<°
*	°>	°>	<°	°>	<°
(	<°	<°	<°	≡°	<°
)	°>	°>		°>	
iden	°>	°>		°>	

5.2.14. Является грамматикой предшествования.

	a	*b*	*c*	**A**	**B**	**D**
a		≡	<•		≡	
b		•>			•>	
c		•>			•>	
A		≡			•>	
B		•>			•>	
D			<•		≡	

5.2.19. Это грамматика операторного предшествования

	(	+	*	*iden*	)
(	<°	<°	<°	<°	≡°
+	<°	°>	<°	<°	°>
*	<°	°>	°>	<°	°>
iden		°>	°>		°>
)		°>	°>		°>

5.2.20. Является грамматикой предшествования. 5.2.2.2 Существуют отношения простого предшествования и функции предшествования *f=(1, 0, 0, 2, 0)*, *g=(0, 1, 1, 1, 0)*.

	S	*a*	*b*	*c*	#
S				≡	
a	≡	<•	<•	<•	
b	≡	<•	<•	<•	
c				•>	•>
#		<•	<•	<•	

5.2.26. Существуют следующие отношения предшествования

	S	*a*	*b*	*c*	#
S	≡	<•	≡	<•	
a	≡	<•		<•	
b		•>	•>	•>	•>
c		•>	•>	•>	•>
#		<•		<•	

5.2.27. Грамматика не является грамматикой простого предшествования: между **S** и *1* слева наблюдается два отношения. 5.2.28. Грамматика не является грамматикой предшествования. 5.2.29. Грамматика не является грамматикой предшествования. 5.3.3. Устранением правой рекурсии получаются необходимые свойства грамматики P={**S**→*0***SA***1*|*0***A***1*; **A**→*1*}. 5.2.33. Отношения предшествования суть следующие

	a	*b*	**S**	**A**	#
#	<•	<•			
a	<•	<•	≡	•>	•>
b	<•	<•	≡	•>	•>
S				≡	
A				•>	•>

5.3.3. Отношения предшествования суть следующие

	S	**A**	*0*	*1*	#
S		≡		<•	
A				≡	
0	≡	≡	<•	<•	
1				•>	•>
#			•>		•>

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Методические указания к выполнению лабораторно - вычислительного практикума по дисциплине “Теоретические основы построения компиляторов” для студентов основного профиля 09.03.02 – “Информационные системы и технологии” всех форм обучения / Разраб. В.Ю. Карлусов, С.А. Кузнецов. – Севастополь: Изд-во СевГУ, 2015. – 36 с.

2. Методические указания к выполнению лабораторных и контрольных работ по дисциплине “Теоретические основы построения компиляторов” для студентов всех форм обучения основного профиля 09.03.02 – “Информационные системы и технологии”. Основные алгоритмы [Текст] / Разраб. В.Ю. Карлусов. – Севастополь: Изд-во СевГУ, 2015. – 44 с.

3. Ахо А. Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ./ А. Ахо, Дж. Ульман. – М.: Мир, 1978. – 619 с.

4. Ахо А. Теория синтаксического анализа, перевода и компиляции. Компиляция / А. Ахо, Дж. Ульман. - М.: Мир, 1978. – 560 с.

5. Вайнгартен Ф. Трансляция языков программирования / Ф. Вайнгартен. – М.: Мир, 1977. – 190 с.

6. Грис Д. Конструирование компиляторов для ЦВМ / Д. Грис. – М.: Мир, 1975. – 544 с.

7. Грогоно П. Программирование на языке Паскаль / П. Грогоно. – М.: Мир, 1982. – 384 с.

8. Льюис Ф. Теоретические основы проектирования компиляторов. / Ф. Льис, Д. Розенкранц, Р. Стирнз. – М.: Мир, 1979. – 564 с.

9. Оллонгрен А. Определение языков программирования интерпретирующими автоматами / А. Оллонгрен - М.: Мир,1972. – 288 с.

10. Рейуорд-Смит В.Дж. Теория формальных языков. Вводный курс / В.Дж. Рейуорд-Смит. – М.: Мир, 1986. – 128 с.

11. Хантер Р. Проектирование и конструирование компиляторов. / Р. Хантер. – М.: Финансы и статистика, 1984. – 232 с.

12. Хопгуд Ф. Методы компиляции. / Ф. Хопгуд. – М.: 1972. – 158 с.

Заказ №		от “		“		20		. Тираж		экз.
				Изд-во СевГУ						