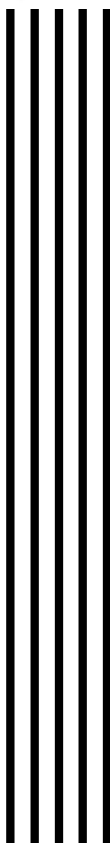


Министерство образования и науки Украины
Севастопольский национальный
технический университет



ФОРМАЛЬНЫЕ ЯЗЫКИ И ГРАММАТИКИ

Методические указания
к расчетно-графическому заданию
по дисциплине «АЛГОРИТМИЗАЦИЯ И
ФОРМАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ»
для студентов специальности
7.091501 «Компьютерные системы и сети»
направления 0915 «Компьютерная инженерия»
дневной формы обучения

Севастополь
2008

УДК 519.682.1+681.142.2

Методические указания к выполнению расчетно-графического задания на тему: «Формальные языки и грамматики» по дисциплине «Алгоритмизация и формальные преобразования» для студентов специальности 7.091501 «Компьютерные системы и сети» направления 0915 «Компьютерная инженерия» дневной формы обучения / Сост. Е.В. Козлова. – Севастополь: Изд-во СевНТУ, 2008. – 20 с.

Целью методических указаний является оказание помощи студентам дневной формы обучения, изучающих дисциплину «Алгоритмизация и формальные преобразования», в овладении практическими навыками проведения анализа и синтеза формальных грамматик при решении задач формализации процессов различной природы.

В методических указаниях представлены основные теоретические положения по следующим разделам: классификация грамматик и языков по Хомскому, соотношения между типами грамматик и типами языков, примеры грамматик и языков, разбор цепочек, а также варианты заданий к выполнению соответствующих расчетных заданий.

Методические указания рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники, протокол № 4 от 27.12.07 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Л.А. Литвинова, кандидат технических наук, доцент

СОДЕРЖАНИЕ

Цель расчетно-графического задания	4
1. Основные теоретические положения	4
1.1 Основные понятия и определения	4
1.2 Классификация грамматик и языков по Хомскому	8
1.3 Соотношения между типами грамматик и типами языков	9
1.4 Примеры грамматик и языков	10
1.5 Разбор цепочек	11
2. Варианты для выполнения расчетного задания	14
3. Форма отчетности	19
4. Контрольные вопросы	20
Библиографический список	20

Цель расчетно-графического задания

Цель расчетно-графического задания: изучение теоретических положений теории формальных языков и грамматик и приобретение навыков решения задач формализации процессов различной природы на основе методов анализа и синтеза формальных языков и соответствующих им грамматик.

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

1.1. Основные понятия и определения

Теория формальных языков и грамматик является основным разделом математической лингвистики – специфической математической дисциплины, ориентированной на изучение структуры естественных и искусственных языков. Эта теория возникла в 50-е годы в работах американского лингвиста Н. Хомского. По характеру используемого математического аппарата теория формальных грамматик и языков близка к теории алгоритмов и к теории автоматов.

Дадим некоторые определения.

Под *грамматикой* понимается некоторая *система правил, задающая множество цепочек (конечных последовательностей) символов языка*.

Эти цепочки можно интерпретировать как языковые объекты разных уровней: словоформы, словосочетания, предложения.

Словоформа или просто слово – это последовательность (цепочка) морфем.

Морфема – это мельчайшая грамматически значимая часть слова.

Например, слово «ведший» состоит из морфем вед+ ш+ий (корень, суффикс, окончание).

Словосочетание или предложение – это цепочка словоформ.

Грамматика языка – это конечное множество правил, определяющих этот язык.

Граматику языка можно рассматривать *теорию структуры этого языка*, то есть теорию повторяющихся закономерностей построения предложений, называемых синтаксической структурой языка.

Синтаксис языка – это правила построения предложений в языке.

Семантика языка – толкование этих правил, правила использования синтаксиса.

Таким образом, иначе можно сказать — **грамматика языка — это конечное множество правил, рекурсивно задающих язык как синтаксическую структуру.**

Первым и основным требованием к грамматике является следующее:

она должна приписывать каждому предложению языка его **структурное описание**, то есть некоторые указания о том, из каких элементов построено предложение, каков их порядок, расположение и т. п.

Структурное описание будет тогда и только тогда однозначно, когда в языке через грамматику однозначно определены его синтаксические единицы и иерархическая взаимосвязь между ними.

Вторым требованием к грамматике является ее **конечность**.

Если допустить грамматики с неопределенным множеством правил, то сама проблема построения грамматик снимется как неразрешимая.

Алфавит — это конечное множество символов.

Предполагается, что термин «символ» имеет достаточно ясный интуитивный смысл и не нуждается в дальнейшем уточнении.

Цепочкой символов в алфавите V называется любая конечная последовательность символов этого алфавита.

Цепочка, которая не содержит ни одного символа, называется **пустой цепочкой**. Для ее обозначения будем использовать символ ε .

Более формально цепочка символов в алфавите V определяется следующим образом:

(1) ε — цепочка в алфавите V ;

(2) если α — цепочка в алфавите V и a — символ этого алфавита, то αa — цепочка в алфавите V ;

(3) β — цепочка в алфавите V тогда и только тогда, когда она является таковой в силу (1) и (2).

Если α и β — цепочки, то цепочка $\alpha\beta$ называется **конкатенацией (или сцеплением)** цепочек α и β .

Например, если $\alpha = ab$ и $\beta = cd$, то $\alpha\beta = abcd$.

Для любой цепочки α всегда $\alpha\varepsilon = \varepsilon\alpha = \alpha$.

Обращением (или реверсом) цепочки α называется цепочка, символы которой записаны в обратном порядке.

Обращение цепочки α будем обозначать α^R .

Например, если $\alpha = abcdef$, то $\alpha^R = fedcba$. Для пустой цепочки: $\varepsilon = \varepsilon^R$.

n -ой степенью цепочки α (будем обозначать α^n) называется конкатенация n цепочек α .

$\alpha^0 = \varepsilon$; $\alpha^n = \alpha\alpha^{n-1} = \alpha^{n-1}\alpha$.

Длина цепочки — это число составляющих ее символов.

Например, если $\alpha = abcdefg$, то длина α равна 7.

Длину цепочки α будем обозначать $|\alpha|$. Длина ε равна 0.

Язык в алфавите V — это подмножество цепочек конечной длины в этом алфавите.

Будем обозначать через V^* множество, содержащее все цепочки в алфавите V , включая пустую цепочку ε .

Например, если $V = \{0,1\}$, то $V^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011 \dots\}$.

Будем обозначать через V^+ множество, содержащее все цепочки в алфавите V , исключая пустую цепочку ε .

Следовательно, $V^* = V^+ \cup \{\varepsilon\}$.

Очевидно, что каждый язык в алфавите V является подмножеством множества V^* .

Известно несколько различных способов описания языков. Один из них использует порождающие грамматики. Именно этот способ описания языков чаще всего будет использоваться нами в дальнейшем.

Декартовым произведением $A \times B$ множеств A и B называется множество $\{(a,b) \mid a \in A, b \in B\}$.

Порождающая грамматика G — это четверка (VT, VN, P, S) , где VT — алфавит *терминальных символов (терминалов)*,

VN — алфавит *нетерминальных символов (нетерминалов)*, не пересекающийся с VT ,

P — конечное подмножество множества $(VT \cup VN)^+ \times (VT \cup VN)^*$; элемент (α, β) множества P называется **правилом вывода** и записывается в виде $\alpha \rightarrow \beta$,

S — **начальный символ (цель)** грамматики, $S \in VN$.

Для записи правил вывода с одинаковыми левыми частями

$\alpha \rightarrow \beta_1 \quad \alpha \rightarrow \beta_2 \quad \dots \quad \alpha \rightarrow \beta_n$

можно пользоваться сокращенной записью $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$.

Каждое β_i , $i = 1, 2, \dots, n$, будем называть **альтернативой** правила вывода из цепочки α .

Пример грамматики: $G_1 = (\{0,1\}, \{A,S\}, P, S)$,

где P состоит из правил $S \rightarrow 0A1$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

Цепочка $\beta \in (VT \cup VN)^*$ **непосредственно выводима** из цепочки $\alpha \in (VT \cup VN)^+$ в грамматике $G = (VT, VN, P, S)$ (обозначим $\alpha \rightarrow \beta$), если $\alpha = \xi_1 \gamma \xi_2$, $\beta = \xi_1 \delta \xi_2$,

где $\xi_1, \xi_2, \delta \in (VT \cup VN)^*$, $\gamma \in (VT \cup VN)^+$ и правило вывода $\gamma \rightarrow \delta$ содержится в P .

Например, цепочка 00A11 непосредственно выводима из 0A1 в грамматике G_1 .

Цепочка $\beta \in (VT \cup VN)^*$ **выводима** из цепочки $\alpha \in (VT \cup VN)^+$ в грамматике $G = (VT, VN, P, S)$ (обозначим $\alpha \Rightarrow \beta$), если существуют цепочки $\gamma_0, \gamma_1, \dots, \gamma_n$ ($n \geq 0$), такие, что $\alpha = \gamma_0 \rightarrow \gamma_1 \rightarrow \dots \rightarrow \gamma_n = \beta$.

Последовательность $\gamma_0, \gamma_1, \dots, \gamma_n$ называется **выводом длины n** .

Например, $S \Rightarrow 000A111$ в грамматике G_1 (см. пример выше), т.к. существует вывод $S \rightarrow 0A1 \rightarrow 00A11 \rightarrow 000A111$. Длина вывода равна 3.

Языком, порождаемым грамматикой $G = (VT, VN, P, S)$, называется множество $L(G) = \{\alpha \in VT^* \mid S \Rightarrow \alpha\}$.

Другими словами, $L(G)$ – это все цепочки в алфавите VT , которые выводимы из S с помощью P .

Например, $L(G_1) = \{0^n 1^n \mid n > 0\}$.

Цепочка $\alpha \in (VT \cup VN)^*$, для которой $S \Rightarrow \alpha$, называется **сентенциальной формой** в грамматике $G = (VT, VN, P, S)$.

Таким образом, язык, порождаемый грамматикой, можно определить как множество терминальных сентенциальных форм.

Грамматики G_1 и G_2 называются **эквивалентными**, если $L(G_1) = L(G_2)$.

Например, $G_1 = (\{0,1\}, \{A,S\}, P_1, S)$ и $G_2 = (\{0,1\}, \{S\}, P_2, S)$

$P_1: S \rightarrow 0A1$

$P_2: S \rightarrow 0S1 \mid 01$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

эквивалентны, т.к. обе порождают язык $L(G_1) = L(G_2) = \{0^n 1^n \mid n > 0\}$.

Грамматики G_1 и G_2 **почти эквивалентны**, если

$L(G_1) \cup \{\varepsilon\} = L(G_2) \cup \{\varepsilon\}$.

Другими словами, грамматик почти эквивалентны, если языки, ими порождаемые, отличаются не более, чем на ε .

Например,

$G_1 = (\{0,1\}, \{A,S\}, P_1, S)$ и $G_2 = (\{0,1\}, \{S\}, P_2, S)$

$P_1: S \rightarrow 0A1$

$P_2: S \rightarrow 0S1 \mid \varepsilon$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

почти эквивалентны, т.к. $L(G_1) = \{0^n 1^n \mid n > 0\}$, а $L(G_2) = \{0^n 1^n \mid n \geq 0\}$, т.е. $L(G_2)$ состоит из всех цепочек языка $L(G_1)$ и пустой цепочки, которая в $L(G_1)$ не входит.

1.2. Классификация грамматик и языков по Хомскому

ТИП 0:

Грамматика $G = (VT, VN, P, S)$ называется *грамматикой типа 0*, если на правила вывода не накладывается никаких ограничений (кроме тех, которые указаны в определении грамматики).

ТИП 1:

Грамматика $G = (VT, VN, P, S)$ называется *неукорачивающей грамматикой*, если каждое правило из P имеет вид $\alpha \rightarrow \beta$, где $\alpha \in (VT \cup VN)^+$, $\beta \in (VT \cup VN)^+$ и $|\alpha| \leq |\beta|$.

Грамматика $G = (VT, VN, P, S)$ называется *контекстно-зависимой (КЗ)*, если каждое правило из P имеет вид $\alpha \rightarrow \beta$,

где $\alpha = \xi_1 A \xi_2$; $\beta = \xi_1 \gamma \xi_2$; $A \in VN$; $\gamma \in (VT \cup VN)^+$; $\xi_1, \xi_2 \in (VT \cup VN)^*$.

Грамматику типа 1 можно определить как неукорачивающую либо как контекстно-зависимую.

Выбор определения не влияет на множество языков, порождаемых грамматиками этого класса, поскольку доказано, что множество языков, порождаемых неукорачивающими грамматиками, совпадает с множеством языков, порождаемых КЗ-грамматиками.

ТИП 2:

Грамматика $G = (VT, VN, P, S)$ называется *контекстно-свободной (КС)*, если каждое правило из P имеет вид $A \rightarrow \beta$, где $A \in VN$, $\beta \in (VT \cup VN)^+$.

Грамматика $G = (VT, VN, P, S)$ называется *укорачивающей контекстно-свободной (УКС)*, если каждое правило из P имеет вид $A \rightarrow \beta$, где $A \in VN$, $\beta \in (VT \cup VN)^*$.

Грамматику типа 2 можно определить как контекстно-свободную либо как укорачивающую контекстно-свободную.

Возможность выбора обусловлена тем, что для каждой УКС-грамматики существует почти эквивалентная КС-грамматика.

ТИП 3:

Грамматика $G = (VT, VN, P, S)$ называется *праволинейной*, если каждое правило из P имеет вид $A \rightarrow tB$ либо $A \rightarrow t$, где $A \in VN$, $B \in VN$, $t \in VT$.

Грамматика $G = (VT, VN, P, S)$ называется *леволинейной*, если каждое правило из P имеет вид $A \rightarrow Bt$ либо $A \rightarrow t$, где $A \in VN$, $B \in VN$, $t \in VT$.

Граматику типа 3 (регулярную, R-грамматику) можно определить как праволинейную либо как леволинейную.

Выбор определения не влияет на множество языков, порождаемых грамматиками этого класса, поскольку доказано, что множество языков, порождаемых праволинейными грамматиками, совпадает с множеством языков, порождаемых леволинейными грамматиками.

1.3. Соотношения между типами грамматик и типами языков

Соотношения между типами грамматик можно описать следующим образом:

1. Любая регулярная грамматика является КС-грамматикой;
2. Любая регулярная грамматика является УКС-грамматикой;
3. Любая КС-грамматика является КЗ-грамматикой;
4. Любая КС-грамматика является неукорачивающей грамматикой;
5. Любая КЗ-грамматика является грамматикой типа 0;
6. Любая неукорачивающая грамматика является грамматикой типа 0.

Замечание1: УКС-грамматика, содержащая правила вида $A \rightarrow \varepsilon$, не является КЗ-грамматикой и не является неукорачивающей грамматикой.

Язык $L(G)$ является **языком типа k** , если его можно описать грамматикой типа k .

Соотношения между типами языков можно описать как:

1. Каждый регулярный язык является КС-языком, но существуют КС-языки, которые не являются регулярными (например, $L = \{a^n b^n \mid n > 0\}$).
2. Каждый КС-язык является КЗ-языком, но существуют КЗ-языки, которые не являются КС-языками (например, $L = \{a^n b^n c^n \mid n > 0\}$).
3. Каждый КЗ-язык является языком типа 0.

Замечание 2: УКС-язык, содержащий пустую цепочку, не является КЗ-языком.

Замечание 3: следует подчеркнуть, что если язык задан грамматикой типа k , то это не значит, что не существует грамматики типа k' ($k' > k$), описывающей тот же язык. Поэтому, когда говорят о языке типа k , обычно имеют в виду максимально возможный номер k .

Например,

КЗ-грамматика
 $G1 = (\{0,1\}, \{A,S\}, P1, S)$ и

КС-грамматика
 $G2 = (\{0,1\}, \{S\}, P2, S),$

P1: $S \rightarrow 0A1$
 $0A \rightarrow 00A1$
 $A \rightarrow \epsilon$

P2: $S \rightarrow 0S1|01$

описывают один и тот же язык $L = L(G1) = L(G2) = \{0^n 1^n \mid n > 0\}$.

Язык L называют КС-языком, т.к. существует КС-грамматика, его описывающая. Но он не является регулярным языком, т.к. не существует регулярной грамматики, описывающей этот язык.

1.4. Примеры грамматик и языков

При описании грамматики указываются правила вывода P . В них большие латинские буквы обозначают нетерминальные символы, S - цель грамматики, все остальные символы – терминальные.

1) Язык типа 0: $L(G) = \{a^2 b^{n^2-1} \mid n \geq 1\}$

G: $S \rightarrow \text{aaCFD}$

$$F \rightarrow AFB \mid AB$$
$$AB \rightarrow bBA$$
$$Ab \rightarrow bA$$
$$AD \rightarrow D$$
$$\text{Cb} \rightarrow \text{bC}$$
$$\text{CB} \rightarrow \text{C}$$
$$\text{bCD} \rightarrow \varepsilon$$

2) Язык типа 1: $L(G) = \{ a^n b^n c^n, n \geq 1 \}$

G: $S \rightarrow aSBC \mid abC$

$$\text{CB} \rightarrow \text{BC}$$
$$bB \rightarrow bb$$
$$bC \rightarrow bc$$
$$cC \rightarrow cc$$

3) Язык типа 2: $L(G) = \{(ac)^n (cb)^n \mid n > 0\}$

G: $S \rightarrow aOb \mid accb$

$$0 \rightarrow cSc$$

4) Язык типа 3: $L(G) = \{\omega \mid \omega \in \{a,b\}^+, \text{ где нет двух рядом стоящих } a\}$

$$G: \quad S \rightarrow A \perp | B \perp$$
$$A \rightarrow a \mid Ba$$
$$B \rightarrow b \mid Bb \mid Ab$$

1.5. Разбор цепочек

Цепочка принадлежит языку, порождаемому грамматикой, только в том случае, если существует ее вывод из цели этой

грамматики. Процесс построения такого вывода (а, следовательно, и определения принадлежности цепочки языку) называется *разбором*.

С практической точки зрения наибольший интерес представляет разбор по **контекстно-свободным (КС и УКС) грамматикам**. Их порождающей мощности достаточно для описания большей части синтаксической структуры языков программирования, для различных подклассов КС-грамматик имеются хорошо разработанные практически приемлемые способы решения задачи разбора.

Рассмотрим основные понятия и определения, связанные с разбором по КС-грамматике.

Вывод цепочки $\beta \in (VT)^*$ из $S \in VN$ в КС-грамматике $G = (VT, VN, P, S)$, называется *левым (левосторонним)*, если в этом выводе каждая очередная сентенциальная форма получается из предыдущей заменой самого левого нетерминального символа.

Вывод цепочки $\beta \in (VT)^*$ из $S \in VN$ в КС-грамматике $G = (VT, VN, P, S)$, называется *правым (правосторонним)*, если в этом выводе каждая очередная сентенциальная форма получается из предыдущей заменой самого правого нетерминального символа.

В грамматике для одной и той же цепочки может быть несколько выводов, эквивалентных в том смысле, что в них в одних и тех же местах применяются одни и те же правила вывода, но в различном порядке.

Например, для цепочки $a+b+a$ в грамматике

$$G = (\{a, b, +\}, \{S, T\}, \{S \rightarrow T \mid T+S; T \rightarrow a \mid b\}, S)$$

можно построить выводы:

- (1) $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow a+T+T \rightarrow a+b+T \rightarrow a+b+a$
- (2) $S \rightarrow T+S \rightarrow a+S \rightarrow a+T+S \rightarrow a+b+S \rightarrow a+b+T \rightarrow a+b+a$
- (3) $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow T+T+a \rightarrow T+b+a \rightarrow a+b+a$

Здесь (2) – левосторонний вывод, (3) – правосторонний, а (1) не является ни левосторонним, ни правосторонним, но все эти выводы являются эквивалентными в указанном выше смысле.

Для КС-грамматик можно ввести удобное графическое представление вывода, называемое деревом вывода, причем для всех эквивалентных выводов дерева вывода совпадают.

Дерево называется *деревом вывода* (или *деревом разбора*) в КС-грамматике $G = \{VT, VN, P, S\}$, если выполнены следующие условия:

1. Каждая вершина дерева помечена символом из множества

$(VN \cup VT \cup \varepsilon)$, при этом корень дерева помечен символом S ; листья – символами из $(VT \cup \varepsilon)$;

2. Если вершина дерева помечена символом $A \in VN$, а ее непосредственные потомки – символами $a_1, a_2, \dots, a_n$,

где каждое $a_i \in (VT \cup VN)$, то $A \rightarrow a_1 a_2 \dots a_n$ – правило вывода в этой грамматике;

3. Если вершина дерева помечена символом $A \in VN$, а ее единственный непосредственный потомок помечен символом ε , то $A \rightarrow \varepsilon$ – правило вывода в этой грамматике.

Пример дерева вывода для цепочки $a+b+a$ в грамматике G показан на рисунке 1.

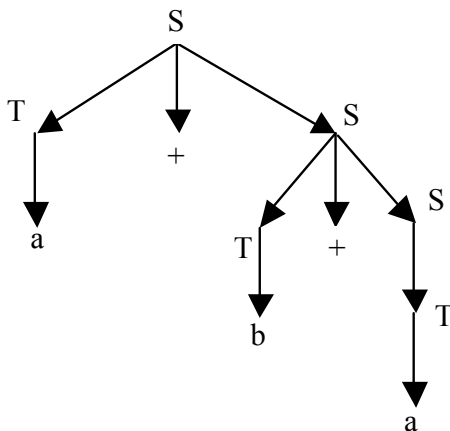


Рисунок 1 – Дерево вывода цепочки в грамматике G

КС-грамматика G называется **неоднозначной**, если существует хотя бы одна цепочка $\alpha \in L(G)$, для которой может быть построено два или более различных деревьев вывода.

Это утверждение эквивалентно тому, что цепочка α имеет два или более разных левосторонних (или правосторонних) выводов.

В противном случае грамматика называется **однозначной**.

Язык, порождаемый грамматикой, называется **неоднозначным**, если он не может быть порожден никакой однозначной грамматикой.

Пример неоднозначной грамматики:

$G = (\{if, then, else, a, b\}, \{S\}, P, S)$,

где $P = \{S \rightarrow if\ b\ then\ S\ else\ S \mid if\ b\ then\ S \mid a\}$.

В этой грамматике для цепочки $if\ b\ then\ if\ b\ then\ a\ else\ a$ можно построить два различных дерева вывода.

Однако это не означает, что язык $L(G)$ обязательно неоднозначный. Определенная нами **неоднозначность** – это **свойство грамматики, а не языка**, т.е. для некоторых неоднозначных грамматик существуют эквивалентные им однозначные грамматики.

Если грамматика используется для определения языка программирования, то она должна быть однозначной.

В приведенном выше примере разные деревья вывода предполагают соответствие `else` разным `then`. Если договориться, что `else` должно соответствовать ближайшему к нему `then`, и подправить грамматику G , то неоднозначность будет устранена:

$$S \rightarrow \text{if } b \text{ then } S \mid \text{if } b \text{ then } S' \text{ else } S \mid a$$

$$S' \rightarrow \text{if } b \text{ then } S' \text{ else } S' \mid a$$

Проблема, порождает ли данная КС-грамматика однозначный язык (т.е. существует ли эквивалентная ей однозначная грамматика), является **алгоритмически неразрешимой**.

Однако можно указать некоторые виды правил вывода, которые приводят к неоднозначности:

$$(1) A \rightarrow AA \mid \alpha$$

$$(2) A \rightarrow A\alpha A \mid \beta$$

$$(3) A \rightarrow \alpha A \mid A\beta \mid \gamma$$

$$(4) A \rightarrow \alpha A \mid \alpha A\beta A \mid \gamma$$

Пример неоднозначного КС-языка: $L = \{a^i b^j c^k \mid i = j \text{ или } j = k\}$.

Интуитивно это объясняется тем, что цепочки с $i = j$ должны порождаться группой правил вывода, отличных от правил, порождающих цепочки с $j = k$. Но тогда, по крайней мере, некоторые из цепочек с $i = j = k$ будут порождаться обеими группами правил и, следовательно, будут иметь по два разных дерева вывода.

Одна из грамматик, порождающих L , такова:

$$S \rightarrow AB \mid DC$$

$$A \rightarrow aA \mid \varepsilon$$

$$B \rightarrow bBc \mid \varepsilon$$

$$C \rightarrow cC \mid \varepsilon$$

$$D \rightarrow aDb \mid \varepsilon$$

Очевидно, что она неоднозначна.

Дерево вывода можно строить **нисходящим** либо **восходящим** способом.

При нисходящем разборе дерево вывода формируется от корня к листьям; на каждом шаге для вершины, помеченной нетерминальным символом, пытаются найти такое правило вывода,

чтобы имеющиеся в нем терминальные символы «проецировались» на символы исходной цепочки.

Метод восходящего разбора заключается в том, что исходную цепочку пытаются «свернуть» к начальному символу S ; на каждом шаге ищут подцепочку, которая совпадает с правой частью какого-либо правила вывода; если такая подцепочка находится, то она заменяется нетерминалом из левой части этого правила.

Если грамматика однозначная, то при любом способе построения будет получено одно и то же дерево разбора.

2. ВАРИАНТЫ ДЛЯ ВЫПОЛНЕНИЯ РАСЧЕТНОГО ЗАДАНИЯ

Расчетное задание включает в себя решение пяти задач.

В индивидуальное задание включаются задачи с номерами:

№ вар	Номера задач	№ вар	Номера задач	№ вар	Номера задач	№ вар	Номера задач
1	1(a), 8, 15, 22, 29	8	8, 12, 16, 20, 24	15	15, 19, 23, 1(b), 2	22	22, 12, 2, 4.2, 6
2	2, 9, 16, 23, 30	9	9, 13, 17, 21, 25	16	16, 20, 3, 3, 4.1, 5	23	23, 13, 3, 20.2, 8
3	3, 10, 17, 24, 4.4	10	10, 14, 18, 22, 26	17	17, 21, 6, 7, 8	24	24, 14, 6, 9, 10
4	4, 3, 11, 18, 25, 1(b)	11	11, 15, 19, 4.9, 27	18	18, 22, 9, 10, 11	25	25, 15, 30, 4.8, 1(a)
5	5, 12, 19, 26, 27	12	12, 16, 24, 20.1, 28	19	19, 23, 12, 13, 14	26	26, 16, 5, 13, 16
6	6, 13, 20, 27, 4.7	13	13, 17, 21, 25, 29	20	24, 15, 16, 17, 20.2	27	27, 17, 18, 3, 4.5
7	7, 14, 21, 4.10, 15	14	14, 18, 22, 26, 4.11	21	21, 25, 19, 2, 4.6	28	28, 20.3, 3, 4.4, 7

Номер варианта выбирается по номеру студента в списке студенческой группы.

Задачи:

1. Дана грамматика. Построить вывод заданной цепочки.

$$a) \quad S \rightarrow T \mid T+S \mid T-S$$

$$T \rightarrow F \mid F*T$$

$$F \rightarrow a \mid b$$

$$\text{Цепочка: } a-b*a+b$$

$$b) S \rightarrow aSBC \mid abC$$

$$CB \rightarrow BC$$

$$bB \rightarrow bb$$

$$bC \rightarrow bc$$

$$cC \rightarrow cc$$

$$\text{Цепочка: } aaabbbcc$$

2. Построить все сентенциальные формы для грамматики с правилами:

$$S \rightarrow A+B \mid B+A$$

$$A \rightarrow a$$

$$B \rightarrow b$$

3. К какому типу по Хомскому относится данная грамматика? Какой язык она порождает? Каков тип языка?

a) $S \rightarrow APA$

$$P \rightarrow + \mid -$$

$$A \rightarrow a \mid b$$

c) $S \rightarrow 1B$

$$B \rightarrow B0 \mid 1$$

b) $S \rightarrow aQb \mid \varepsilon$

$$Q \rightarrow cSc$$

d) $S \rightarrow A \mid SA \mid SB$

$$A \rightarrow a$$

$$B \rightarrow b$$

4. Построить грамматику, порождающую язык:

4.1. $L = \{a^n b^m \mid n, m \geq 1\}$

4.2. $L = \{\alpha\beta\gamma\epsilon \mid \alpha, \beta, \gamma - \text{любые цепочки из } a \text{ и } b\}$

4.3. $L = \{a_1 a_2 \dots a_n a_n \dots a_2 a_1 \mid a_i = 0 \text{ или } 1, n \geq 1\}$

4.4. $L = \{a^n b^m \mid n \neq m; n, m \geq 0\}$

4.5. $L = \{\text{цепочки из } 0 \text{ и } 1 \text{ с неравным числом } 0 \text{ и } 1\}$

4.6. $L = \{\alpha\alpha \mid \alpha \in \{a, b\}^+\}$

4.7. $L = \{\omega \mid \omega \in \{0, 1\}^+ \text{ и содержит равное количество } 0 \text{ и } 1, \text{ причем любая подцепочка, взятая с левого конца, содержит единиц не меньше, чем нулей}\}.$

4.8. $L = \{(a^{2m} b^m)^n \mid m \geq 1, n \geq 0\}$

4.9. $L = \{a^{3^n+1} \mid n \geq 1\}$

4.10. $L = \{a^{n^2} \mid n \geq 1\}$

4.11. $L = \{a^{n^3+1} \mid n \geq 1\}$

5. К какому типу по Хомскому относится данная грамматика? Какой язык она порождает? Каков тип языка?

a) $S \rightarrow a \mid Ba$

$$B \rightarrow Bb \mid b$$

b) $S \rightarrow Ab$

$$A \rightarrow Aa \mid ba$$

c) $S \rightarrow 0A1 \mid 01$

$$0A \rightarrow 00A1$$

$$A \rightarrow 01$$

d) $S \rightarrow AB$

$$AB \rightarrow BA$$

$$A \rightarrow a$$

$$B \rightarrow b$$

e) $S \rightarrow A \mid B$

f) $S \rightarrow 0A \mid 1S$

$$\begin{aligned} A &\rightarrow aAb \mid 0 \\ B &\rightarrow aBbb \mid 1 \end{aligned}$$

$$\begin{aligned} A &\rightarrow 0A \mid 1B \\ B &\rightarrow 0B \mid 1B \mid \perp \end{aligned}$$

$$\begin{aligned} \text{g)} \quad S &\rightarrow 0S \mid S0 \mid D \\ D &\rightarrow DD \mid 1A \mid \varepsilon \\ A &\rightarrow 0B \mid \varepsilon \\ B &\rightarrow 0A \mid 0 \end{aligned}$$

$$\begin{aligned} \text{h)} \quad S &\rightarrow 0A \mid 1S \mid \varepsilon \\ A &\rightarrow 1A \mid 0B \\ B &\rightarrow 0S \mid 1B \end{aligned}$$

$$\begin{aligned} \text{i)} \quad S &\rightarrow SS \mid A \\ A &\rightarrow a \mid bb \end{aligned}$$

$$\begin{aligned} \text{j)} \quad S &\rightarrow AB\perp \\ A &\rightarrow a \mid cA \\ B &\rightarrow bA \end{aligned}$$

$$\begin{aligned} \text{k)} \quad S &\rightarrow aBA \mid \varepsilon \\ B &\rightarrow bSA \\ AA &\rightarrow c \end{aligned}$$

$$\begin{aligned} \text{l)} \quad S &\rightarrow Ab \mid c \\ A &\rightarrow Ba \\ B &\rightarrow cS \end{aligned}$$

6. Эквивалентны ли грамматики с правилами:

$$\begin{aligned} \text{6.1} \quad S &\rightarrow AB \text{ и} & S &\rightarrow AS \mid SB \mid AB \\ A &\rightarrow a \mid Aa & A &\rightarrow a \\ B &\rightarrow b \mid Bb & B &\rightarrow b \end{aligned}$$

$$\begin{aligned} \text{6.2} \quad S &\rightarrow aSL \mid aL \text{ и} & S &\rightarrow aSBc \mid abc \\ L &\rightarrow Kc & cB &\rightarrow Bc \\ cK &\rightarrow Kc & bB &\rightarrow bb \\ K &\rightarrow b \end{aligned}$$

7. Построить КС-грамматику, эквивалентную грамматике с правилами:

$$\begin{aligned} \text{a)} \quad S &\rightarrow aAb \\ aA &\rightarrow aaAb \\ A &\rightarrow \varepsilon \\ \text{b)} \quad S &\rightarrow AB \mid ABS \\ AB &\rightarrow BA \\ BA &\rightarrow AB \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

8. Построить регулярную грамматику, эквивалентную грамматике с правилами:

$$\begin{aligned} \text{a)} \quad S &\rightarrow A \mid AS \\ A &\rightarrow a \mid bb \\ \text{b)} \quad S &\rightarrow A.A \\ A &\rightarrow B \mid BA \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

9. Доказать, что грамматика с правилами:

$$\begin{aligned} S &\rightarrow aSBC \mid abC \\ CB &\rightarrow BC \end{aligned}$$

$$\begin{aligned} bB &\rightarrow bb \\ cC &\rightarrow cc \end{aligned}$$

порождает язык $L = \{a^n b^n c^n \mid n \geq 1\}$. Для этого показать, что в данной грамматике

- 1) выводится любая цепочка вида $a^n b^n c^n$ ($n \geq 1$) и
- 2) не выводятся никакие другие цепочки.

10. Дана грамматика с правилами:

$$S \rightarrow S0 \mid S1 \mid D0 \mid D1$$

$$D \rightarrow H.$$

$$H \rightarrow 0 \mid 1 \mid H0 \mid H1$$

Построить восходящим методом дерево вывода для цепочки:

10.1001

11. Дана грамматика с правилами:

$$S \rightarrow \text{if } B \text{ then } S \mid B = E$$

$$E \rightarrow B \mid B + E$$

$$B \rightarrow a \mid b$$

Построить нисходящим методом дерево вывода для цепочки:

if a then b = a+b+b

12. Определить тип грамматики. Описать язык, порождаемый этой грамматикой. Написать для этого языка КС-грамматику.

$$S \rightarrow P \perp$$

$$P \rightarrow 1P1 \mid 0P0 \mid T$$

$$T \rightarrow 021 \mid 120R$$

$$R1 \rightarrow 0R$$

$$R0 \rightarrow 1$$

$$R \perp \rightarrow 1 \perp$$

13. Построить регулярную грамматику, порождающую цепочки в алфавите $\{a, b\}$, в которых символ a не встречается два раза подряд.

14. Построить грамматику, порождающую сбалансированные относительно круглых скобок цепочки в алфавите $\{a, (,), \perp\}$. Сбалансированную цепочку α определим рекуррентно: цепочка α сбалансирована, если

a) α не содержит скобок,

b) $\alpha = (\alpha_1)$ или $\alpha = \alpha_1 \alpha_2$, где α_1 и α_2 сбалансированы.

15. Написать КС-грамматику, порождающую язык L , и вывод для цепочки $aacbbbcsaa$ в этой грамматике.

$$L = \{a^n cb^m ca^n \mid n, m > 0\}.$$

16. Написать КС-грамматику для описания десятичных целочисленных констант.

17. Дана грамматика G . Определить ее тип; язык, порождаемый этой грамматикой; тип языка.

$$G: \{S \rightarrow 0A1,$$

$$0A \rightarrow 00A1,$$

$$A \rightarrow \varepsilon\}$$

18. Дан язык $L = \{1^{3n+2} 0^n \mid n \geq 0\}$. Определить его тип, написать грамматику, порождающую L . Построить левосторонний и правосторонний выводы, дерево разбора для цепочки 1111111100.

19. Привести пример грамматики, все правила которой имеют вид $A \rightarrow Bt$, либо $A \rightarrow tB$, либо $A \rightarrow t$, для которой не существует эквивалентной ей регулярной грамматики.

20. Написать общие алгоритмы построения по данным КС-грамматикам G_1 и G_2 , порождающим языки L_1 и L_2 , КС-грамматики для:

$$20.1. L = L_1 \cup L_2$$

$$20.2 L = L_1 * L_2$$

$$20.3 L = L_1^*$$

Замечание: $L = L_1 * L_2$ – это конкатенация языков L_1 и L_2 , т.е. $L = \{\alpha\beta \mid \alpha \in L_1, \beta \in L_2\}$; $L = L_1^*$ – это итерация языка L_1 ,

т.е. объединение $\{\varepsilon\} \cup L_1 \cup L_1 * L_1 \cup L_1 * L_1 * L_1 \cup \dots$

21. Написать КС-грамматику для описания десятичных вещественных констант с фиксированной точкой.

22. Показать, что грамматика $E \rightarrow E+E \mid E * E \mid (E) \mid i$ неоднозначна. Как описать этот же язык с помощью однозначной грамматики?

23. Показать, что наличие в КС-грамматике правил вида

$$a) A \rightarrow AA \mid \alpha$$

$$b) A \rightarrow A\alpha A \mid \beta$$

$$c) A \rightarrow \alpha A \mid A\beta \mid \gamma$$

где $\alpha, \beta, \gamma \in (VT \cup VN)^*$, $A \in VN$, делает ее неоднозначной. Можно ли преобразовать эти правила таким образом, чтобы полученная эквивалентная грамматика была однозначной?

24. Показать, что грамматика G неоднозначна. Какой язык она порождает? Является ли этот язык однозначным?

$$G: \{S \rightarrow aAc \mid aB, B \rightarrow bc, A \rightarrow b\}$$

25. Дана КС-грамматика $G = \{VT, VN, P, S\}$. Предложить алгоритм построения множества $X = \{A \in VN \mid A \Rightarrow \varepsilon\}$.

26. Для произвольной КС-грамматики G предложить алгоритм, определяющий, пуст ли язык $L(G)$.

27. Написать приведенную грамматику, эквивалентную данной.

$$27.1 S \rightarrow aABS \mid bCACd$$

$$A \rightarrow bAB \mid cSA \mid cCC$$

$$B \rightarrow bAB \mid cSB$$

$$\begin{aligned}
C &\rightarrow cS \mid c \\
27.2 \quad S &\rightarrow aAB \mid E \\
A &\rightarrow dDA \mid \varepsilon \\
B &\rightarrow bE \mid f \\
C &\rightarrow cAB \mid dSD \mid a \\
D &\rightarrow Ea \\
E &\rightarrow fA \mid g
\end{aligned}$$

28. Язык называется распознаваемым, если существует алгоритм, который за конечное число шагов позволяет получить ответ о принадлежности любой цепочки языку. Если число шагов зависит от длины цепочки и может быть оценено до выполнения алгоритма, язык называется легко распознаваемым. Доказать, что язык, порождаемый неукорачивающей грамматикой, легко распознаваем.

29. Доказать, что любой конечный язык, в который не входит пустая цепочка, является регулярным языком.

30. Доказать, что нециклическая КС-грамматика порождает конечный язык.

Замечание: Нетерминальный символ $A \in VN$ – циклический, если в грамматике существует вывод $A \Rightarrow \xi_1 A \xi_2$. КС-грамматика называется циклической, если в ней имеется хотя бы один циклический символ.

3. ФОРМА ОТЧЕТНОСТИ

Расчетное задание в виде пояснительной записки представляется к защите. Пояснительная записка оформляется на листах формата А4 на одной стороне каждого листа и содержит следующее: титульный лист, оформленный в соответствии с требованиями высшей школы, содержание, постановка задачи, ход выполнения работы, анализ результатов, заключение, библиографический список.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие основные понятия теории формальных языков и грамматик Вы знаете?
2. Перечислите основные классы формальных грамматик.
3. Как по виду правил подстановки определить тип грамматики по Хомскому?
4. Какие формальные языки считаются эквивалентными?
5. Если два формальных языка эквивалентны, то означает ли это эквивалентность соответствующих им грамматик?
6. Поясните понятие «почти эквивалентные грамматики».
7. Что означает однозначность или неоднозначность грамматики?
8. Какие существуют приемы, позволяющие снять неоднозначность грамматики?
9. Дайте определение порождающей грамматики. Перечислите компоненты её описания.
10. Какие методы построения деревьев разбора Вы знаете? В чем их отличия?
11. Какие методы преобразования грамматик Вы знаете?
12. Перечислите основные свойства формальных языков на основании теорем Поста и Хомского.
13. В чем Вы видите связь теории формальных языков и грамматик с теорией конечных автоматов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Волкова И.А. Формальные грамматики и языки. Элементы теории трансляции / И.А. Волкова, Т.В. Руденко. – 2-е изд., перераб. и доп. – М.: МГУ, 1999. – 149с.
2. Левитин А. Алгоритмы. Введение в разработку и анализ / А. Левитин. – М.: Издательский дом «Вильямс», 2006. – 576с.
3. Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход / М.В. Мозговой. – СПб.: Наука и техника, 2006. – 320с.
4. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М.: МЦНМО, 2001. – 960с.

Заказ № ____ от « ____ » _____ 2008 г. Тираж ____ экз.
Изд-во СевНТУ