

Министерство образования и науки Украины
Севастопольский национальный технический университет



МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторной работы № 5
«Формальные грамматики и их свойства»
по дисциплине «Алгоритмизация и формальные преобразования»
для студентов направления подготовки
«Компьютерная инженерия»
дневной формы обучения

Севастополь
2014

УДК 004.421

Методические указания к выполнению лабораторной работы «Формальные грамматики и их свойства» по дисциплине «Алгоритмизация и формальные преобразования» для студентов направления подготовки «Компьютерная инженерия» дневной формы обучения. / Сост. Козлова Е.В. – Севастополь: Изд-во СевНТУ, 2014. – 16с.

Целью методических указаний является оказание помощи студентам направления подготовки «Компьютерная инженерия» дневной формы обучения, в изучении основ теории формальных языков и грамматик с целью их применения к решению задач проектирования системного программного обеспечения.

В настоящих методических указаниях представлены некоторые теоретические положения по следующим разделам дисциплины: «Основные понятия теории формальных языков и грамматик», «Классификация грамматик», «Сентенциальные грамматики», а также варианты индивидуальных заданий к выполнению лабораторной работы и примеры решения типовых задач.

Методические рекомендации рассмотрены и утверждены на заседании кафедры информационных технологий и компьютерных систем, протокол № 11 от 25 июня 2014 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензенты: Фисун С.Н. к. т. н., доцент,
Щепин Ю.Н., к. т. н., доцент.

СОДЕРЖАНИЕ

Введение	3
1. Краткие теоретические сведения	3
1.1. Основные понятия	3
1.2. Классификация грамматик	7
1.3. Цепочки вывода	9
1.4. Сентенциальная форма грамматики	10
2. Варианты индивидуальных заданий	13
Библиографический список	15

Введение

Целью выполнения лабораторной работы является освоение студентами основ анализа и синтеза формальных грамматик, применяемых для решения инженерных задач, овладение навыками распознавания типов формальных языков и грамматик по классификации Хомского, формирование умения построения грамматики по заданному описанию формального языка, а также получения сентенциальных форм по заданной грамматике.

Номер варианта индивидуального задания для выполнения лабораторной работы соответствует номеру фамилии студента в списке учебной группы.

Отчёт о выполнении лабораторной работы должен быть выполнен **аккуратно**, желательно в виде компьютерного набора. Однако не запрещается и рукописный текст, можно выполнять рисунки и схемы вручную, но **обязательно** с использованием линейки.

На титульном листе **обязательно** должен быть указан номер выполняемого варианта. Образец титульного листа приведен в Приложении. Отчёт о выполнении работы должен включать следующие разделы:

1. Содержание.
2. Цель работы.
3. Постановка задачи и описание хода её решения (для каждого пункта индивидуального задания)

Работы, выполненные небрежно или не для своего варианта задания будут возвращены без проверки!!!

1. Краткие теоретические сведения

1.1. Основные понятия

Тексты на любом языке (естественном или формальном) представляют собой цепочки символов некоторого алфавита.

Цепочкой символов называют произвольную упорядоченную конечную последовательность символов, записанных один за другим. Понятие *символа* является базовым в теории формальных языков. Для цепочки символов имеют значение три фактора: состав входящих в цепочку символов, их количество и порядок символов в цепочке. Далее цепочки символов будем обозначать греческими буквами: α , β , γ и др.

Цепочки символов α и β *равны* ($\alpha = \beta$), если они имеют один и тот же состав символов, одно и то же их количество и одинаковый порядок следования символов в цепочке. Количество символов в цепочке определяет её *длину*. Длина цепочки символов α обозначается как $|\alpha|$.

Можно выделить следующие операции над цепочками символов:

– *конкатенация* – объединение, сложение двух цепочек – это дописывание второй цепочки в конец первой. Конкатенация цепочек α и β обозначается как $\alpha\beta$.

Например: $\alpha = аб$, $\beta = вг$, тогда $\alpha\beta = абвг$. При этом $\alpha\beta \neq \beta\alpha$, так как в цепочке важен порядок символов. Но конкатенация обладает свойством ассоциативности: $(\alpha\beta)\gamma = \alpha(\beta\gamma)$.

– *замена (подстановка)* – замена подцепочки символов на любую произвольную цепочку символов. В результате получается новая цепочка символов.

Например, $\gamma = абвг$. Разобьём эту цепочку символов на подцепочки: $\alpha = а$, $\omega = б$, $\beta = вг$ и выполним подстановку цепочки $\upsilon = аба$ вместо подцепочки ω .

Получим новую цепочку $\gamma' = аабавг$. Таким образом, подставка выполняется путём разбиения исходной цепочки на подцепочки и конкатенации.

– *обращение* – запись символов цепочки в обратном порядке. Эта операция обозначается как α^R . Если $\alpha = абвг$, то $\alpha^R = гвба$. Для операции обращения справедливо следующее:

$$(\alpha\beta)^R = \beta^R\alpha^R.$$

– *итерация* – повторение цепочки n раз, где $n > 0$ – это конкатенация цепочки с собой n раз, обозначается как α^n . Если $n = 0$, то результатом итерации будет пустая цепочка символов.

Пустая цепочка символов – это цепочка, не содержащая ни одного символа. Будем обозначать такую цепочку как ε . Для пустой цепочки справедливы следующие равенства:

$$\begin{aligned} |\varepsilon| &= 0 \\ \varepsilon\alpha &= \alpha\varepsilon = \alpha \\ \varepsilon^R &= \varepsilon \\ \varepsilon^n &= \varepsilon, \varepsilon \geq 0 \\ \alpha^0 &= \varepsilon \end{aligned}$$

Основой любого языка является алфавит, определяющий набор допустимых символов языка.

Алфавит – это счётное множество допустимых символов языка. Будем обозначать это множество как V .

Цепочка символов a является *цепочкой над алфавитом V* : $a(V)$, если в нее входят только символы, принадлежащие множеству символов V .

Для любого алфавита V пустая цепочка может, как являться, так и не являться цепочкой над алфавитом.

Если V – некоторый алфавит, то

V^+ – множество всех цепочек над алфавитом V без пустой цепочки;

V^* – множество всех цепочек над алфавитом V , включая пустую цепочку.

Языком L над алфавитом V ($L(V)$) называется некоторое счетное подмножество цепочек конечной длины из множества всех цепочек над алфавитом V . Множество цепочек языка не обязано быть конечным; хотя каждая цепочка символов, входящая в язык, обязана иметь конечную длину, эта длина может быть сколь угодно большой и формально ничем не ограничена.

Цепочку символов, принадлежащую заданному языку, часто называют *предложением* языка, а множество цепочек символов некоторого языка $L(V)$ — множеством предложений этого языка.

Кроме алфавита язык предусматривает также правила построения допустимых цепочек, поскольку обычно далеко не все цепочки над заданным алфавитом принадлежат языку. Символы могут объединяться в слова или лексемы — элементарные конструкции языка, на их основе строятся предложения — более сложные конструкции. И те и другие в общем виде являются цепочками символов, но предусматривают некоторые правила построения.

Таким образом, необходимо указать эти правила, или, строго говоря, задать язык.

В общем случае язык можно определить тремя способами:

1) перечислением всех допустимых цепочек языка;

- 2) указанием способа порождения цепочек языка (заданием грамматики языка);
- 3) определением метода распознавания цепочек языка.

Первый из методов является чисто формальным и на практике не применяется, так как большинство языков содержат бесконечное число допустимых цепочек и перечислить их просто невозможно. Иногда для чисто формальных языков можно перечислить множество входящих в них цепочек, прибегнув к математическим определениям множеств. Однако этот подход уже стоит ближе ко второму способу.

Например, запись $L(\{0,1\}) = \{0^n 1^n, n > 0\}$ задает язык над алфавитом $V = \{0,1\}$, содержащий все последовательности с чередующимися символами 0 и 1, начинающиеся с 0 и заканчивающиеся 1. Видно, что пустая цепочка символов в этот язык не входит.

Если изменить условие в этом определении с $n > 0$ на $n \geq 0$, то получим почти эквивалентный язык $L'(\{0,1\})$, содержащий пустую цепочку.

Второй способ предусматривает некоторое описание правил, с помощью которых строятся цепочки языка. Тогда любая цепочка, построенная с помощью этих правил из символов алфавита языка, будет принадлежать заданному языку.

Третий способ предусматривает построение некоторого логического устройства (распознавателя) — автомата, который на входе получает цепочку символов, а на выходе выдает ответ, принадлежит или нет эта цепочка заданному языку.

Грамматика — это описание способа построения предложений некоторого языка. Иными словами, грамматика — это математическая система, определяющая язык. Она относится ко второму способу определения языков — порождению цепочек символов.

Граматику языка можно описать различными способами. Например, можно использовать формальное описание грамматики, построенное на основе системы правил (или продукций).

Правило (или продукция) — это упорядоченная пара цепочек символов (α, β) . В правилах важен порядок цепочек, поэтому их чаще записывают в виде

$$\alpha \rightarrow \beta \text{ (или } \alpha ::= \beta \text{)}.$$

Такая запись читается как « α порождает β » или « α по определению есть β ».

Грамматика языка программирования содержит правила двух типов: первые (определяющие синтаксические конструкции языка) довольно легко поддаются формальному описанию; вторые (определяющие семантические ограничения языка) обычно излагаются в неформальной форме.

Поэтому любое описание (или стандарт) языка программирования обычно состоит из двух частей: вначале формально излагаются правила построения синтаксических конструкции, а потом на естественном языке дается описание семантических правил.

Язык, заданный грамматикой G , обозначается как $L(G)$. Две грамматики, G и G' , называются эквивалентными, если они определяют один и тот же язык: $L(G) = L(G')$.

Две грамматики, G и G' , называются почти эквивалентными, если заданные ими языки различаются не более чем на пустую цепочку символов: $L(G) \setminus \{\varepsilon\} = L(G') \setminus \{\varepsilon\}$.

Формально грамматика G определяется как четверка $G(VT, VN, P, S)$, где VT — множество терминальных символов, или алфавит терминальных символов; VN — множество нетерминальных символов, или алфавит нетерминальных символов; P — множество правил (продукций) грамматики вида $\alpha \rightarrow \beta$, где $\alpha \in (VN \cup VT)^+$, $\beta \in (VN \cup VT)^*$; S — целевой (начальный) символ грамматики $S \in VN$.

Алфавиты терминальных и нетерминальных символов грамматики не пересекаются:

$$VN \cap VT = \emptyset.$$

Это значит, что каждый символ в грамматике может быть либо терминальным, либо нетерминальным, но не может быть терминальным и нетерминальным одновременно.

Целевой символ грамматики — это всегда нетерминальный символ.

Множество $V = V_N \cup V_T$ называют полным алфавитом грамматики G .

Множество терминальных символов V_T содержит символы, которые входят в алфавит языка, порожденного грамматикой. Как правило, символы из множества V_T встречаются только в цепочках правых частей правил.

Множество нетерминальных символов V_N содержит символы, которые определяют слова, понятия, конструкции языка. Каждый символ этого множества может встречаться в цепочках как левой, так и правой частей правил грамматики, но он обязан хотя бы один раз быть в левой части хотя бы одного правила. Правила грамматики строятся так, чтобы в левой части каждого правила был хотя бы один нетерминальный символ.

Во множестве правил грамматики может быть несколько правил, имеющих одинаковые левые части вида, $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$.

Тогда эти правила объединяют вместе и записывают в виде:

$$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n.$$

Одной строке в такой записи соответствует сразу n правил.

Такую форму записи правил грамматики называют формой Бэкуса-Наура. Форма Бэкуса-Наура, как правило, предусматривает также, что нетерминальные символы берутся в угловые скобки: $\langle \rangle$.

Иногда знак $\rightarrow$ в правилах грамматики заменяют на знак $::=$.

Пример грамматики, которая определяет язык целых десятичных чисел со знаком:

$G(\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +\}, \{\langle \text{число} \rangle, \langle \text{чс} \rangle, \langle \text{цифра} \rangle\}, P, \langle \text{число} \rangle)$

P :

$\langle \text{число} \rangle \rightarrow \langle \text{чс} \rangle \mid +\langle \text{чс} \rangle \mid -\langle \text{чс} \rangle$

$\langle \text{чс} \rangle \rightarrow \langle \text{цифра} \rangle \mid \langle \text{чс} \rangle \langle \text{цифра} \rangle$

$\langle \text{цифра} \rangle \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Рассмотрим составляющие элементы грамматики G :

- множество терминальных символов V_T содержит двенадцать элементов: десять десятичных цифр и два знака;
- множество нетерминальных символов V_N содержит три элемента: символы $\langle \text{число} \rangle$, $\langle \text{чс} \rangle$ и $\langle \text{цифра} \rangle$;
- множество правил содержит 15 правил, которые записаны в три строки (то есть имеется только три различные правые части правил);
- целевым символом грамматики является символ $\langle \text{число} \rangle$.

Та же самая грамматика для языка целых десятичных чисел со знаком, в которой нетерминальные символы обозначены большими латинскими буквами (далее это будет часто применяться в примерах) имеет вид:

$G'(\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +\}, \{S, T, F\}, P, S)$

P :

$S \rightarrow T \mid +T \mid -T$;

$T \rightarrow F \mid TF$;

$F \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Здесь изменилось только множество нетерминальных символов.

Теперь $V_N = \{S, T, F\}$. Язык, заданный грамматикой, не изменился — можно сказать, что грамматики G и G' эквивалентны.

Особенность рассмотренных выше формальных грамматик в том, что они позволяют определить бесконечное множество цепочек языка с помощью конечного набора правил. В такой

форме записи грамматики возможность пользоваться конечным набором правил достигается за счет рекурсивных правил.

Рекурсия в правилах грамматики выражается в том, что один из нетерминальных символов определяется сам через себя. Рекурсия может быть непосредственной (явной) – тогда символ определяется сам через себя в одном правиле, либо косвенной (неявной) – тогда это же самое происходит через цепочку правил.

В рассмотренной выше грамматике G непосредственная рекурсия присутствует в правиле $\langle \text{чс} \rangle \rightarrow \langle \text{чс} \rangle \langle \text{цифра} \rangle$, а в эквивалентной ей грамматике G' — в правиле $T \rightarrow TF$.

Чтобы рекурсия не была бесконечной, для участвующего в ней нетерминального символа грамматики должны существовать также и другие правила, которые определяют его, минуя самого себя, и позволяют избежать бесконечного рекурсивного определения.

Таковыми правилами являются $\langle \text{чс} \rangle \rightarrow \langle \text{цифра} \rangle$ — в грамматике G и $T \rightarrow F$ – в грамматике G' .

1.2 Классификация грамматик

Согласно классификации, предложенной американским лингвистом Ноамом Хомским, профессором Массачусетского технологического института, формальные грамматики классифицируются по структуре их правил. Если все без исключения правила грамматики удовлетворяют некоторой заданной структуре, то такую грамматику относят к определенному типу. Достаточно иметь в грамматике одно правило, не удовлетворяющее требованиям структуры правил, и она уже не попадает в заданный тип. По классификации Хомского выделяют четыре типа грамматик.

Тип 0: грамматики с фразовой структурой

На структуру их правил не накладывается никаких ограничений: для грамматики вида $G(VT, VN, P, S)$, $V = VN \cup VT$, правила имеют вид $\alpha \rightarrow \beta$, где $\alpha \in V^+$, $\beta \in V^*$.

Это самый общий тип грамматик. В него попадают все без исключения формальные грамматики, но часть из них может быть также отнесена и к другим классификационным типам. Грамматики, которые относятся только к типу 0 и не могут быть отнесены к другим типам, являются самыми сложными по структуре. Практического применения грамматики, относящиеся только к типу 0, не имеют.

Тип 1: контекстно-зависимые (КЗ) и неукорачивающие грамматики

В этот тип входят два основных класса грамматик:

- Контекстно-зависимые грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, имеют правила вида

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2, \text{ где } \alpha_1, \alpha_2 \in V^*, A \in VN, \beta \in V^+.$$

- Неукорачивающие грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, имеют правила вида:

$$\alpha \rightarrow \beta, \text{ где } \alpha, \beta \in V^+, |\beta| \geq |\alpha|.$$

Структура правил КЗ-грамматик такова, что при построении предложений заданного ими языка один и тот же нетерминальный символ может быть заменен на ту или иную цепочку символов в зависимости от того контекста, в котором он встречается. Именно поэтому эти грамматики называют *контекстно-зависимыми* (КЗ).

Цепочки α_1 и α_2 в правилах грамматики обозначают контекст (α_1 — левый контекст, а α_2 — правый контекст), в общем случае любая из них (или даже обе) может быть пустой. Говоря иными словами, значение одного и того же символа может быть различным в зависимости от того, в каком контексте он встречается.

Неукорачивающие грамматики имеют такую структуру правил, что при построении предложений языка, заданного грамматикой, любая цепочка символов может быть заменена на

цепочку символов не меньшей длины. Отсюда и название *неукорачивающие*. Доказано, что эти два класса грамматик эквивалентны.

Это значит, что для любого языка, заданного КЗ-грамматикой, можно построить неукорачивающую грамматику, которая будет задавать эквивалентный язык, и наоборот: для любого языка, заданного неукорачивающей грамматикой, можно построить КЗ-грамматику, которая будет задавать эквивалентный язык.

Тип 2: контекстно-свободные (КС) грамматики

Неукорачивающие контекстно-свободные (НКС) грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, имеют правила вида:

$$A \rightarrow \beta, \text{ где } A \in VN, \beta \in V^+.$$

Такие грамматики называют НКС-грамматиками, поскольку видно, что в правой части правил у них должен всегда стоять как минимум один символ.

Существует также почти эквивалентный им класс грамматик — укорачивающие контекстно-свободные (УКС) грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, правила которых могут иметь вид:

$$A \rightarrow \beta, \text{ где } A \in VN, \beta \in V^*.$$

Эти два класса составляют тип контекстно-свободных (КС) грамматик. Разница между ними заключается лишь в том, что в УКС-грамматиках в правой части правил может присутствовать пустая цепочка, а в НКС-грамматиках — нет. Отсюда ясно, что язык, заданный НКС-грамматикой, не может содержать пустой цепочки.

Тип 3: регулярные грамматики

К типу регулярных относятся два эквивалентных класса грамматик: левوليнейные и праволинейные.

Левوليнейные грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, могут иметь правила двух видов:

$$A \rightarrow B\gamma \text{ или } A \rightarrow \gamma, \text{ где } A, B \in VN, \gamma \in VT^*.$$

В свою очередь, праволинейные грамматики $G(VT, VN, P, S)$, $V = VN \cup VT$, могут иметь правила тоже двух видов:

$$A \rightarrow \gamma B \text{ или } A \rightarrow \gamma, \text{ где } A, B \in VN, \gamma \in VT^*.$$

Языки классифицируются в соответствии с типами грамматик, с помощью которых они заданы. Причем, поскольку один и тот же язык в общем случае может быть задан сколь угодно большим количеством грамматик, которые могут относиться к различным классификационным типам, для классификации самого языка среди всех его грамматик выбирается грамматика с максимально возможным классификационным типом. Например, если язык L может быть задан грамматиками G_1 и G_2 , относящимися к типу 1 (КЗ), грамматикой G_3 , относящейся к типу 2 (КС), и грамматикой G_4 , относящейся к типу 3 (регулярные), сам язык должен быть отнесен к типу 3 и является регулярным языком.

1.3 Цепочки вывода

Выводом называется процесс порождения предложения языка на основе правил определяющей язык грамматики.

Цепочка $\beta = \delta_1 \gamma \delta_2$ называется непосредственно выводимой из цепочки $\alpha = \delta_1 \omega \delta_2$ в грамматике $G(VT, VN, P, S)$, $V = VN \cup VT$, $\delta_1, \gamma, \delta_2 \in V^*$, $\omega \in V^+$, если в грамматике существует правило: $\omega \rightarrow \gamma \in P$.

Иными словами, цепочка β выводима из цепочки α в том случае, если можно взять несколько символов в цепочке α , поменять их на другие символы, согласно некоторому правилу грамматики, и получить цепочку β .

Непосредственная выводимость цепочки β из цепочки α обозначается как: $\alpha \Rightarrow \beta$.

Цепочка β называется выводимой из цепочки α ($\alpha \Rightarrow^* \beta$) в случае, если выполняется одно из двух условий:

- β непосредственно выводима из α ($\alpha \Rightarrow \beta$);
- существует γ такая, что γ выводима из α , и β непосредственно выводима из γ , т.е.
 $(\alpha \Rightarrow^* \gamma, \gamma \Rightarrow \beta)$.

Пример 1. Грамматика для языка целых десятичных чисел со знаком:

$G(\{0,1,2,3,4,5,6,7,8,9,-,+\}, \{S, T, F\}, P, S)$

P :

$S \rightarrow T \mid +T \mid -T$;

$T \rightarrow F \mid TF$;

$F \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$.

Построим несколько цепочек вывода в этой грамматике:

1. $S \Rightarrow -T \Rightarrow -TF \Rightarrow -TFF \Rightarrow -FFF \Rightarrow -4FF \Rightarrow -47F \Rightarrow -479$
2. $S \Rightarrow T \Rightarrow TF \Rightarrow T8 \Rightarrow F8 \Rightarrow 18$
3. $T \Rightarrow TF \Rightarrow T0 \Rightarrow TF0 \Rightarrow T50 \Rightarrow F50 \Rightarrow 350$
4. $F \Rightarrow 5$

Получаем, что: $S \Rightarrow^* -479$, $S \Rightarrow^* 18$, $T \Rightarrow^* 350$, $F \Rightarrow^* 5$.

Пример 2. Задана грамматика $G(\{a,b,c\}, \{B,C,D,S\}, P, S)$ с правилами

P :

$S \rightarrow BD$

$B \rightarrow aBbC \mid ab$

$Cb \rightarrow bC$

$CD \rightarrow Dc$

$bDc \rightarrow bcc$

$abD \rightarrow abc$

Эта грамматика задает язык $L(G) = \{a^n b^n c^n \mid n > 0\}$.

Пример вывода предложения $aaaabbbbccccc$ для этого языка на основе грамматики G выглядит следующим образом (для понимания каждого шага вывода подцепочка, для которой выполняется подстановка, выделена жирным шрифтом):

$S \Rightarrow \mathbf{BD} \Rightarrow \mathbf{aBbCD} \Rightarrow \mathbf{aaBbCbCD} \Rightarrow \mathbf{aaaBbCbCbCD} \Rightarrow \mathbf{aaaabbCbCbCD} \Rightarrow \mathbf{aaaabbbCCbCD} \Rightarrow$
 $\Rightarrow \mathbf{aaaabbbbCbCCD} \Rightarrow \mathbf{aaaabbbbCCCD} \Rightarrow \mathbf{aaaabbbbCCDc} \Rightarrow \mathbf{aaaabbbbCDcc}$
 $\Rightarrow \mathbf{aaaabbbbDccc} \Rightarrow \mathbf{aaaabbbbccccc}$

Таким образом, цепочка $aaaabbbbccccc$ выводима из S :

$S \Rightarrow^* aaaabbbbccccc$.

1.4 Сентенциальная форма грамматики

Вывод называется *законченным* (или *конечным*), если на основе цепочки β , полученной в результате этого вывода, нельзя больше сделать ни одного шага вывода.

Иначе говоря, вывод называется законченным, если цепочка β , полученная в результате этого вывода, пустая или содержит только терминальные символы грамматики $G(VT, VN, P, S)$: $\beta \in VT^*$. Цепочка β , полученная в результате законченного вывода, называется *конечной* цепочкой вывода.

В рассмотренном выше примере 1 все построенные выводы являются законченными, вывод $S \Rightarrow^* -4FF$ (из первой цепочки в примере) будет незаконченным.

Цепочка символов $\alpha \in V^*$ называется *сентенциальной формой* грамматики $G(VT, VN, P, S)$, $V = VT \cup VN$, если она выводима из целевого символа грамматики S : $S \Rightarrow^* \alpha$.

Если цепочка $\alpha \in VT^*$ получена в результате законченного вывода, то она называется *конечной сентенциальной формой*.

Из рассмотренного выше примера можно заключить, что цепочки символов -479 и 18 являются конечными сентенциальными формами грамматики целых десятичных чисел со знаком, так как существуют выводы $S \Rightarrow^* -479$ и $S \Rightarrow^* 18$ (выводы 1 и 2). Цепочка $F8$ из вывода 2, например, тоже является сентенциальной формой, поскольку справедливо $S \Rightarrow^* F8$, но она не является конечной сентенциальной формой вывода. В то же время в выводах 3 и 4 примера 1 явно не присутствуют сентенциальные формы.

Левосторонний и правосторонний выводы

Вывод называется *левосторонним*, если в нем на каждом шаге вывода правило грамматики применяется всегда к крайнему левому нетерминальному символу в цепочке. Другими словами, если на каждом шаге вывода происходит подстановка цепочки символов на основании правила грамматики вместо крайнего левого нетерминального символа в исходной цепочке.

Аналогично, вывод называется *правосторонним*, если в нем на каждом шаге вывода правило грамматики применяется всегда к крайнему правому нетерминальному символу в цепочке.

Если рассмотреть цепочки вывода из примера 1, то в нем выводы 1 и 4 являются левосторонними, выводы 2, 3 и 4 — правосторонними (вывод 4 является одновременно и правосторонним, и левосторонним).

Для грамматик типов 2 и 3 (КС-грамматик и регулярных грамматик) для любой сентенциальной формы всегда можно построить левосторонний или правосторонний вывод. Для грамматик других типов это не всегда возможно, так как по структуре их правил не всегда можно выполнить замену крайнего левого или крайнего правого нетерминального символа в цепочке.

Рассмотренный в примере 2 вывод $S \Rightarrow^* aaaabbbbccccc$ для грамматики G , задающей язык $L(G) = \{a^n b^n c^n \mid n > 0\}$, не является ни левосторонним, ни правосторонним. Грамматика относится к типу 1, и в данном случае для нее нельзя построить такой вывод, на каждом шаге которого только один нетерминальный символ заменялся бы на цепочку символов.

Дерево вывода

Деревом вывода грамматики $G(VT, VN, P, S)$ называется дерево (граф), которое соответствует некоторой цепочке вывода и удовлетворяет следующим условиям:

- каждая вершина дерева обозначается символом грамматики $A \in (VT \cup VN \cup \{\epsilon\})$;
- корнем дерева является вершина, обозначенная целевым символом грамматики — S ;
- листьями дерева (концевыми вершинами) являются вершины, обозначенные терминальными символами грамматики или символом пустой цепочки ϵ ;

- если некоторый узел дерева обозначен нетерминальным символом $A \in VN$, а связанные с ним узлы — символами $b_1, b_2 \dots b_n$; $n > 0$, $\forall n \geq i > 0: b_i \in (VT \cup VN \cup \{\epsilon\})$, то в грамматике $G(VT, VN, P, S)$ существует правило $A \rightarrow b_1, b_2 \dots b_n \in P$.

Из определения видно, что по структуре правил дерево вывода в указанном виде всегда можно построить только для грамматик типов 2 и 3 (контекстно-свободных и регулярных). Для грамматик других типов дерево вывода в таком виде можно построить не всегда (либо же оно будет иметь несколько иной вид).

На основе примера 1 построим деревья вывода для цепочек выводов 1 и 2. Эти деревья приведены на рисунке 1.

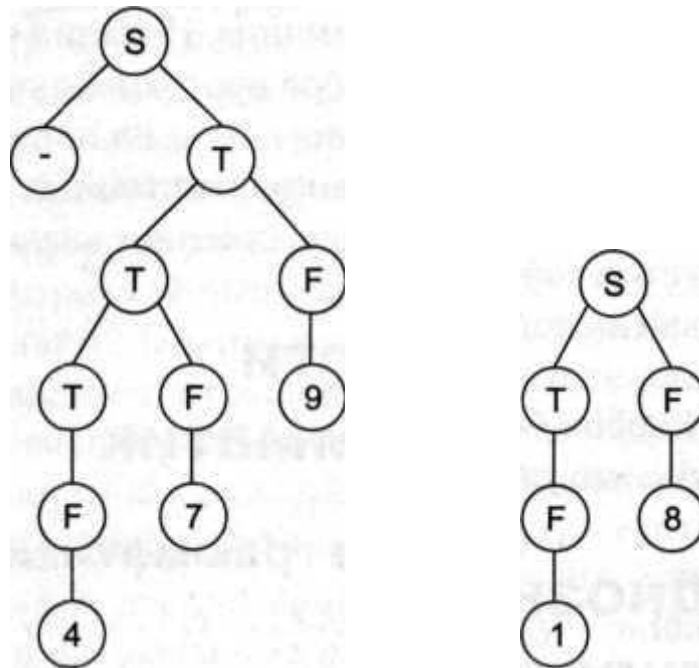


Рисунок 1. Примеры деревьев вывода для грамматики целых десятичных чисел со знаком

Для того чтобы построить дерево вывода, достаточно иметь только цепочку вывода. Дерево вывода можно построить двумя способами: сверху вниз и снизу вверх. Для строго формализованного построения дерева вывода всегда удобнее пользоваться строго определенным выводом: либо левосторонним, либо правосторонним.

При построении дерева вывода сверху вниз построение начинается с целевого символа грамматики, который помещается в корень дерева. Затем в грамматике выбирается необходимое правило, и на первом шаге вывода корневой символ раскрывается на несколько символов первого уровня. На втором шаге среди всех концевых вершин дерева выбирается крайняя (крайняя левая — для левостороннего вывода, крайняя правая — для правостороннего) вершина, обозначенная нетерминальным символом, для этой вершины выбирается нужное правило грамматики, и она раскрывается на несколько вершин следующего уровня. Построение дерева заканчивается, когда все концевые вершины обозначены терминальными символами, в противном случае надо вернуться ко второму шагу и продолжить построение.

Построение дерева вывода снизу вверх начинается с листьев дерева. В качестве листьев выбираются терминальные символы конечной цепочки вывода, которые на первом шаге построения образуют последний уровень (слой) дерева. Построение дерева идет по слоям. На

втором шаге построения в грамматике выбирается правило, правая часть которого соответствует крайним символам в слое дерева (крайним правым символам при правостороннем выводе и крайним левым — при левостороннем). Выбранные вершины слоя соединяются с новой вершиной, которая выбирается из левой части правила. Новая вершина попадает в слой дерева вместо выбранных вершин. Построение дерева закончено, если достигнута корневая вершина (обозначенная целевым символом), а иначе надо вернуться ко второму шагу и повторить его над полученным слоем дерева.

Если для каждой цепочки символов языка, заданного грамматикой, можно построить единственный левосторонний (и единственный правосторонний) вывод или, что то же самое, построить единственное дерево вывода, то такая грамматика называется *однозначной*. Иначе грамматика называется *неоднозначной*.

Пример 3. Рассмотрим грамматику $G(\{+, -, *, /, (,), a, b\}, \{S\}, P, S)$:

P:

$$S \rightarrow S+S \mid S-S \mid S \times S \mid S/S \mid (S) \mid a \mid b$$

Возьмем цепочку $a \times b + a$ и построим для нее левосторонний вывод. Получится два варианта:

$$S \Rightarrow S+S \Rightarrow S \times S+S \Rightarrow a \times S+S \Rightarrow a \times b+S \Rightarrow a \times b+a,$$

$$S \Rightarrow S \times S \Rightarrow a \times S \Rightarrow a \times S+S \Rightarrow a \times b+S \Rightarrow a \times b+a.$$

Каждому из этих вариантов будет соответствовать свое дерево вывода (рисунок 2).

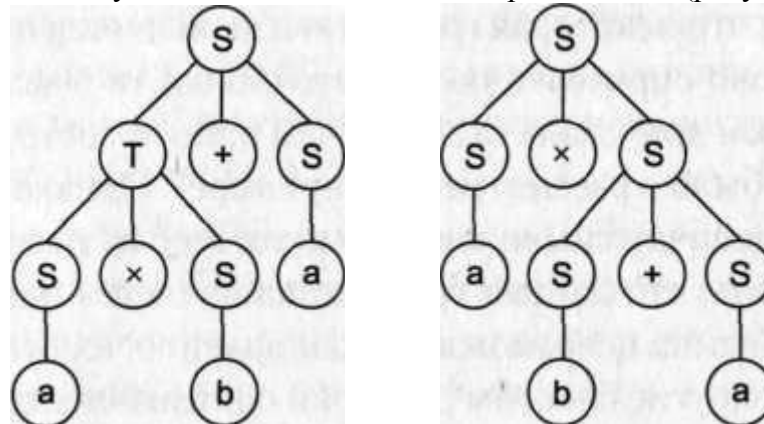


Рисунок 2. — Два варианта дерева вывода цепочки $a \times b + a$ для неоднозначной грамматики арифметических выражений

Для КС-грамматик существуют определенного вида правила, по наличию которых во всем множестве правил грамматики можно утверждать, что она является неоднозначной:

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. $A \rightarrow \alpha A \mid \alpha A\beta A \mid \gamma$

Здесь $A \in VN$; $\alpha, \beta, \gamma \in (VN \cup VT)^*$.

Если в заданной грамматике встречается хотя бы одно правило подобного вида, то такая грамматика точно будет неоднозначной. Однако отсутствие подобных правил во всем множестве правил грамматики не означает, что данная грамматика является однозначной.

Для грамматики из примера 3 можно показать, что она является неоднозначной, поскольку она содержит правила второго вида: $S \rightarrow S + S \mid a$.

2. Индивидуальные задания к выполнению лабораторной работы

Задание 1. Определить тип (по классификации Хомского) заданной грамматики.

№ варианта	Описание правил грамматики
1	$S \rightarrow abba, S \rightarrow baa$
2	$S \rightarrow aS, S \rightarrow bS, S \rightarrow \varepsilon$
3	$S \rightarrow abSba, S \rightarrow a$
4	$S \rightarrow AD, A \rightarrow aA, A \rightarrow \varepsilon, D \rightarrow bDc, D \rightarrow \varepsilon$
5	$K \rightarrow KT, T \rightarrow Ba, KB \rightarrow B, B \rightarrow b$
6	$S \rightarrow aSa, S \rightarrow T, T \rightarrow Tb, T \rightarrow \varepsilon$
7	$F \rightarrow aFb, F \rightarrow bF a, F \rightarrow FF, F \rightarrow c$
8	$S \rightarrow TaV, T \rightarrow bV, V \rightarrow cV c, V \rightarrow \varepsilon, ac \rightarrow c$
9	$S \rightarrow a \mid B, aB \rightarrow Bb \mid b$
10	$S \rightarrow Ab, A \rightarrow Aa \mid ba$
11	$S \rightarrow 0A1 \mid 01, 0A \rightarrow 00A1, A \rightarrow 01$
12	$S \rightarrow AB, AB \rightarrow BA, A \rightarrow a, B \rightarrow b$
13	$F \rightarrow aFb, F \rightarrow ab.$
14	$S \rightarrow AD, A \rightarrow aA, A \rightarrow \varepsilon, D \rightarrow bDc, D \rightarrow \varepsilon.$
15	$K \rightarrow \varepsilon, K \rightarrow a, K \rightarrow b, K \rightarrow aKa, K \rightarrow bKb.$
16	$F \rightarrow KK, K \rightarrow aKb, K \rightarrow ab.$
17	$K \rightarrow BKG, B \rightarrow b, G \rightarrow bb, K \rightarrow ab.$
18	$K \rightarrow KA, KAA \rightarrow AKb, AKA \rightarrow b, A \rightarrow a.$
19	$F \rightarrow AFHA, F \rightarrow AbA, A \rightarrow a, bH \rightarrow bb, AH \rightarrow HA.$
20	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRc, cH \rightarrow Hc, R \rightarrow bc$

Задание 2. Построить все, если возможно, или несколько сентенциальных форм для грамматики с заданными правилами

№ варианта	Описание правил грамматики
1	$F \rightarrow AFHA, F \rightarrow AbA, A \rightarrow a, bH \rightarrow bb, AH \rightarrow HA$
2	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRc, cH \rightarrow Hc, R \rightarrow bc$
3	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRG, GH \rightarrow HG, R \rightarrow bG, G \rightarrow c$
4	$F \rightarrow aFH, F \rightarrow abc, bH \rightarrow bbc, cH \rightarrow Hc$
5	$S \rightarrow aT bc, Tb \rightarrow bT, Tc \rightarrow Ubcc, bU \rightarrow Ub, aU \rightarrow aaT, aU \rightarrow aa.$
6	$S \rightarrow aUa, Tb \rightarrow bT, Ta \rightarrow Ub aa, bU \rightarrow Ub, aU \rightarrow aaT, aU \rightarrow ab$
7	$S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc$
8	$S \rightarrow aSA, S \rightarrow aT, TA \rightarrow bT a, aA \rightarrow Aa, T \rightarrow ba$
9	$S \rightarrow TaS, S \rightarrow U, Ta \rightarrow aaT, TU \rightarrow U, U \rightarrow a$
10	$S \rightarrow A+B \mid B+A, A \rightarrow a, B \rightarrow b$
11	$S \rightarrow aTS, S \rightarrow U, Ta \rightarrow aaT, TU \rightarrow U, U \rightarrow a$
12	$S \rightarrow KT, T \rightarrow aTB, T \rightarrow \varepsilon, aB \rightarrow Baa, KB \rightarrow Ka, K \rightarrow a$
13	$F \rightarrow aFb, F \rightarrow ab.$
14	$S \rightarrow AD, A \rightarrow aA, A \rightarrow \varepsilon, D \rightarrow bDc, D \rightarrow \varepsilon.$
15	$K \rightarrow \varepsilon, K \rightarrow a, K \rightarrow b, K \rightarrow aKa, K \rightarrow bKb.$

16	$F \rightarrow KK, K \rightarrow aKb, K \rightarrow ab.$
17	$K \rightarrow BKG, B \rightarrow b, G \rightarrow bb, K \rightarrow ab.$
18	$K \rightarrow KA, KAA \rightarrow AKb, AKA \rightarrow b, A \rightarrow a.$
19	$F \rightarrow AFHA, F \rightarrow AbA, A \rightarrow a, bH \rightarrow bb, AH \rightarrow HA.$
20	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRc, cH \rightarrow Hc, R \rightarrow bc$

Задание 3. Определить эквивалентность грамматик G и G' , заданных правилами:

№ варианта	Правила грамматики G	Правила грамматики G'
1	$S \rightarrow ab, S \rightarrow aKSb, K \rightarrow bSb, KS \rightarrow b, K \rightarrow \varepsilon$	$S \rightarrow aTb, T \rightarrow \varepsilon, T \rightarrow b, T \rightarrow S, T \rightarrow bSbS$
2	$S \rightarrow aD, D \rightarrow bba, D \rightarrow aDb, D \rightarrow baDa, D \rightarrow aDaDa$	$S \rightarrow aaE, S \rightarrow abJ, E \rightarrow aEJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow abJa, J \rightarrow ba$
3	$S \rightarrow aD, D \rightarrow bba, D \rightarrow baDa, D \rightarrow aDaDa$	$S \rightarrow aaE, S \rightarrow abJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow abJa, J \rightarrow ba$
4	$Q \rightarrow aR, R \rightarrow bba, R \rightarrow aRb, R \rightarrow aRaRa$	$Q \rightarrow aaE, Q \rightarrow abJ, E \rightarrow aEJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow ba$
5	$S \rightarrow aSKMa, S \rightarrow bM, aK \rightarrow Ka, aM \rightarrow Ma, MK \rightarrow KM, bK \rightarrow bb, bM \rightarrow bc, cM \rightarrow cc$	$S \rightarrow aKS Ma, S \rightarrow bc, aM \rightarrow Ma, Ka \rightarrow aK, Kb \rightarrow bb, cM \rightarrow cc$
6	$S \rightarrow aSKMa, S \rightarrow bM, aK \rightarrow Ka, aM \rightarrow Ma, MK \rightarrow KM, bK \rightarrow bb, bM \rightarrow bc, cM \rightarrow cc$	$S \rightarrow aKS Ma, S \rightarrow bc, aM \rightarrow Ma, Ka \rightarrow aK, Kb \rightarrow bb, cM \rightarrow cc$
7	$F \rightarrow ab, F \rightarrow aKFb, K \rightarrow bFb, KF \rightarrow b, K \rightarrow \varepsilon$	$S \rightarrow aTb, T \rightarrow \varepsilon, T \rightarrow b, T \rightarrow S, T \rightarrow bSbS$
8	$R \rightarrow ab, R \rightarrow aKRb, K \rightarrow bRb, KR \rightarrow b, K \rightarrow \varepsilon$	$Q \rightarrow aTb, T \rightarrow \varepsilon, T \rightarrow b, T \rightarrow Q, T \rightarrow bQbQ$
9	$F \rightarrow aC, C \rightarrow bba, C \rightarrow baCa, C \rightarrow aCaCa$	$F \rightarrow aaE, F \rightarrow abJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow abJa, J \rightarrow ba$
10	$R \rightarrow aK, K \rightarrow bba, K \rightarrow aKb, K \rightarrow aKaKa$	$R \rightarrow aaE, R \rightarrow abJ, E \rightarrow aEJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow ba$
11	$R \rightarrow fRKMf, R \rightarrow bM, fK \rightarrow Kf, fM \rightarrow Mf, MK \rightarrow KM, bK \rightarrow bb, bM \rightarrow bc, cM \rightarrow cc$	$R \rightarrow fKRMf, R \rightarrow bc, fM \rightarrow Mf, Kf \rightarrow fK, Kb \rightarrow bb, cM \rightarrow cc$
12	$S \rightarrow aD, D \rightarrow bba, D \rightarrow aDb, D \rightarrow aDaDa$	$S \rightarrow aaE, S \rightarrow abJ, E \rightarrow aEJ, E \rightarrow bJJ, J \rightarrow aaEa, J \rightarrow ba$
13	$R \rightarrow am, R \rightarrow aKRm, K \rightarrow mRm, KR \rightarrow m, K \rightarrow \varepsilon$	$S \rightarrow aTm, T \rightarrow \varepsilon, T \rightarrow m, T \rightarrow S, T \rightarrow mSmS$
14	$S \rightarrow aR, R \rightarrow bba, R \rightarrow aRb, R \rightarrow baRa, R \rightarrow aRaRa$	$S \rightarrow aaG, S \rightarrow abJ, G \rightarrow aGJ, G \rightarrow bJJ, J \rightarrow aaGa, J \rightarrow abJa, J \rightarrow ba$
15	$Q \rightarrow ab, Q \rightarrow aKQb, K \rightarrow bQb, KQ \rightarrow b, K \rightarrow \varepsilon$	$P \rightarrow aTb, T \rightarrow \varepsilon, T \rightarrow b, T \rightarrow P, T \rightarrow bPbP$
16	$M \rightarrow ab, M \rightarrow aKMb, K \rightarrow bMb, KM \rightarrow b, K \rightarrow \varepsilon$	$Q \rightarrow aYb, Y \rightarrow \varepsilon, Y \rightarrow b, Y \rightarrow Q, Y \rightarrow bQbQ$
17	$Y \rightarrow aR, R \rightarrow bba, R \rightarrow aRb, R \rightarrow baRa, R \rightarrow aRaRa$	$Y \rightarrow aaZ, Y \rightarrow abJ, Z \rightarrow aZJ, Z \rightarrow bJJ, J \rightarrow aaZa, J \rightarrow abJa, J \rightarrow ba$
18	$S \rightarrow zp, S \rightarrow zKSp, K \rightarrow pSp, KS \rightarrow p, K \rightarrow \varepsilon$	$S \rightarrow zTp, T \rightarrow \varepsilon, T \rightarrow p, T \rightarrow S, T \rightarrow pSpS$

19	$S \rightarrow aD, D \rightarrow ppa, D \rightarrow paDa,$ $D \rightarrow aDaDa$	$S \rightarrow aaE, S \rightarrow apG, E \rightarrow pGG,$ $G \rightarrow aaEa, G \rightarrow apGa, G \rightarrow pa$
20	$F \rightarrow ab, F \rightarrow aZFb, Z \rightarrow bFb, ZF \rightarrow b, Z \rightarrow \varepsilon$	$S \rightarrow aPb, P \rightarrow \varepsilon, P \rightarrow b, P \rightarrow S,$ $P \rightarrow bSbS$

Задание 4. Описать язык, порождаемый грамматикой:

№ варианта	Описание правил грамматики
1	$F \rightarrow aFb, F \rightarrow ab.$
2	$S \rightarrow AD, A \rightarrow aA, A \rightarrow \varepsilon, D \rightarrow bDc, D \rightarrow \varepsilon.$
3	$K \rightarrow \varepsilon, K \rightarrow a, K \rightarrow b, K \rightarrow aKa, K \rightarrow bKb.$
4	$F \rightarrow KK, K \rightarrow aKb, K \rightarrow ab.$
5	$K \rightarrow BKG, B \rightarrow b, G \rightarrow bb, K \rightarrow ab.$
6	$K \rightarrow KA, KAA \rightarrow AKb, AKA \rightarrow b, A \rightarrow a.$
7	$F \rightarrow AFHA, F \rightarrow AbA, A \rightarrow a, bH \rightarrow bb, AH \rightarrow HA.$
8	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRc, cH \rightarrow Hc, R \rightarrow bc$
9	$F \rightarrow aFH, F \rightarrow aR, RH \rightarrow bRG, GH \rightarrow HG, R \rightarrow bG, G \rightarrow c.$
10	$F \rightarrow aFH, F \rightarrow abc, bH \rightarrow bbc, cH \rightarrow Hc.$
11	$S \rightarrow aTbc, Tb \rightarrow bT, Tc \rightarrow Ubcc, bU \rightarrow Ub, aU \rightarrow aaT, aU \rightarrow aa.$
12	$S \rightarrow aUa, Tb \rightarrow bT, Ta \rightarrow Ub aa, bU \rightarrow Ub, aU \rightarrow aaT, aU \rightarrow ab$
13	$S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc.$
14	$S \rightarrow aSA, S \rightarrow aT, TA \rightarrow bT a, aA \rightarrow Aa, T \rightarrow ba.$
15	$F \rightarrow GJ, G \rightarrow KGT, G \rightarrow KT, KT \rightarrow aTK, Ka \rightarrow aK, Ta \rightarrow aT, KJ \rightarrow Ja, TJ \rightarrow Ca, TC \rightarrow Ca, C \rightarrow a.$
16	$S \rightarrow TaS, S \rightarrow U, Ta \rightarrow aaT, TU \rightarrow U, U \rightarrow a.$
17	$S \rightarrow aTS, S \rightarrow U, Ta \rightarrow aaT, TU \rightarrow U, U \rightarrow a.$
18	$S \rightarrow KT, T \rightarrow aTB, T \rightarrow \varepsilon, aB \rightarrow Baa, KB \rightarrow Ka, K \rightarrow a.$
19	$S \rightarrow BE, B \rightarrow bB, B \rightarrow \varepsilon, E \rightarrow cEd, E \rightarrow \varepsilon.$
20	$F \rightarrow PP, P \rightarrow aPb, P \rightarrow ab.$

Библиографический список

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. 3-е изд. / А.Ю. Молчанов – СПб.: Питер, 2010. – 400с.
2. Ахо А.В. Компиляторы: принципы, технологии и инструментарий, 2-е изд.: Пер. с англ. / А.В. Ахо, М.С. Лам, Р. Сети, Дж. Д. Ульман – М.: Вильямс, 2008. – 1184 с.
3. Карпов Ю.Г. Теория и технология программирования. Основы построения трансляторов / Ю.Г. Карпов– СПб.: БХВ-Петербург, 2005. – 272с.
4. Волкова И.А., Руденко Т.В. Формальные грамматики и языки. Элементы теории трансляции: Учебное пособие. / И.А. Волкова, Т.В. Руденко – Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 1999. – 62с.
5. Языки программирования и методы трансляции: Учебное пособие. / С.З. Свердлов— СПб.: Питер, 2007. — 638с.