МГТУ им. Н.Э.Баумана
Факультет «Фундаментальные науки»
Кафедра «Высшая математика»

Мастихина А.А.

Формальные языки и автоматы

под редакцией Р.С.Исмагилова

Электронное учебное издание
Методические указания к выполнению домашнего задания по дискретной математике

Москва

УДК 518.5

Рецензент:
- кандидат физ.-мат. наук Алексей Иванович Белоусов

Мастихина А.А.
Формальные языки и автоматы: Методические указания к выполнению домашнего задания по дискретной математике. – М, МГТУ им. Н.Э Баумана, 2011. – 23 с.

Изложены основные теоретические сведения из теории формальных языков и конечных автоматов. Разобраны примеры решения типовых задач. Содержит условия типового расчета.

Для студентов, изучающих дискретную математику.

Одобрено учебно-методической комиссией НУК "Фундаментальные науки"МГТУ им. Н.Э.Баумана

Мастихина Анна Антоновна
ФОРМАЛЬНЫЕ ЯЗЫКИ И АВТОМАТЫ:
МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ ДОМАШНЕГО ЗАДАНИЯ ПО ДИСКРЕТНОЙ
МАТЕМАТИКЕ

1 Предварительные сведения.

Данное пособие опирается на некоторые сведения из теории графов. Введем необходимые определения.

Ориентированный псевдограф G — это пара множеств (V, E), где V — множество вершин, E — множество ребер, причем каждому ребру $e \in E$ соответствует упорядоченная пара вершин $(v, u), v, u \in V$ (ребро ведет из v в u). Возможны петли ($v = u$) и кратные ребра (несколько ребер для одной пары вершин). Геометрически ориентированный псевдограф изображается так: для каждой вершины изображается точка на плоскости, а для каждого ребра (v, u) - линия, направленная от вершины v к вершине u, причем разным ребрам соответствуют разные линии.

Путь в ориентированном псевдографе — последовательность ребер $e_1, e_2, ..., e_k$, таких что ребро e_i ведет от v_i к v_{i+1}. Говорят, что такой путь ведет из v_1 в v_{k+1}.

Вершина v достижима из вершины u, если существует путь, ведущий u из в v.

2 Регулярные языки.

Возьмем некоторое конечное множество символов A, назовем его *алфавитом*, а его элементы — *буквами*.

Словом в данном алфавите называется конечная цепочка букв этого алфавита.

Буквы будем обозначать $a, a_1, a_2, ..., b, b_1, ...$, а слова — $\alpha, \beta, ...$, причем $\alpha = a(1)a(2)...a(n)$, где $\alpha(i)$ — i-тая буква слова α.

Длиной слова α называется число букв в данном слове: $|\alpha| = n$. Например, $|abbbc| = 5$.

Введем также пустое слово Λ как слово нулевой длины: $|\Lambda| = 0$.

Слово β называется *подсловом* слова α, если найдутся слова α_1 и α_2, необязательно непустые, что $\alpha = \alpha_1\beta\alpha_2$. Например, подсловами слова abc является само abc, а также a, b, c, ab и bc.

Множество всех возможных слов в алфавите A обозначим A^*.

Языком в данном алфавите A называется любое подмножество L множества всех слов A^*, $L \subset A^*$.

Пример. $A = \{a, b, c\}$

$L = \{\Lambda, aa, abc, cb, bc\}$. A^* — все слова, которые можно составить из букв a, b, c: $\Lambda, a, b, c, aa, ab, ac, ba, ...$

Операции над языками. Регулярные языки.

Рассмотрим произвольный алфавит A и всевозможные языки в нем. Определим следующие операции.

1) *Объединением* языков L_1 и L_2 называется множество слов, входящих хотя бы в один из этих языков: $L = L_1 \vee L_2 = \{\alpha | \alpha \in L_1$ или $\alpha \in L_2\}$.

2) *Конкатенацией (произведением)* языков L_1 и L_2 называется множество слов вида $L_1 \cdot L_2 = \{\alpha\beta$, где $\alpha \in L_1, \beta \in L_2\}$. Таким образом, это слова, получающиеся приписыванием к каждому слову из L_1 слова из L_2. Конкатенация слов α и β есть слово $\alpha\beta$.

Например, пусть $L_1 = \{a, ab, b\}, L_2 = \{b, ca\}$. Тогда

$L_1 \vee L_2 = \{a, ab, b, ca\}, L_1 \cdot L_2 = \{ab, abb, bb, aca, abca, bca\}$

В частности, $\underbrace{L \cdot ... \cdot L}_{k}$ обозначается как L^k и есть $\{\alpha_1 ... \alpha_k | \alpha_i \in L, i = 1...k\}$.

Например, $L = \{a, bb\}, L^2 = \{aa, abb, bba, bbbb\}$

Рассмотрим произвольный язык L и пустое слово Λ. По определению $\Lambda \cdot L = L$, $L \cdot \Lambda = L$.

В качестве языка можно рассматривать и пустое множество слов. Выполнено:

$L \cdot \emptyset = \emptyset = \emptyset \cdot L$

$L \vee \emptyset = L$

3) *Итерацией* языка L называется язык вида $\Lambda \vee L \vee L^2 \vee ... \vee L^i \vee ...$, он обозначается L^*.

Например, в алфавите $A = \{a, b\}$ итерация языка $L = \{a^2, ab\}$ будет $L^* = \{\Lambda, a^2, ab, a^4, abab, a^3b, ab^2a, a^6, a^5b, aba^4, ...\}$.

Для $L = \{a^2\}$ итерация такова: $L^* = \{\Lambda, a^2, a^4, a^6, ...\}$.

Множество всех слов в алфавите $A = \{a_1, ..., a_r\}$ получается итерацией объединения его букв: $A^* = (a_1 \vee ... \vee a_r)^*$.

Язык называется *регулярным*, если его можно получить из простейших языков $\{\Lambda\}, \{a\}, a \in A$, с помощью этих трех операций за конечное число шагов.

Формальное определение.

1) $\emptyset$ — регулярный язык, $\{\Lambda\}$ — регулярный язык, $\{a\}, a \in A$, — регулярный язык.

2) Пусть L_1, L_2 — регулярные языки. Тогда языки $L_1 \cdot L_2$, $L_1 \vee L_2$ и L_1^* также регулярны.

3) Других регулярных языков нет.

Выражение, задающее регулярный язык, называется *регулярным выражением.* Для простейших языков эти выражения — $\emptyset, \Lambda, a$, остальные составляются из простейших с помощью $\vee, \cdot, ^*$ и скобок.

Задача. Составить регулярное выражение для языка в алфавите $\{a, b, c\}$, состоящее из всех слов, начинающихся на ab, но не заканчивающихся на c.

Как уже было сказано, множество всех слов в алфавите $A = \{a, b, c\}$ есть $A^* = (a \vee b \vee c)^*$.

Все слова, начинающиеся на ab — конкатенация ab со множеством всех слов. Выражение для такого языка есть $ab(a \vee b \vee c)^*$. Слово не заканчивается на букву c, значит, оно заканчивается на a или на b.

Поэтому регулярное выражение для данного языка имеет вид $ab(a \vee b \vee c)^*(a \vee b)$.

Задача. Составить регулярное выражение для языка в алфавите $\{a, b, c\}$ из всех слов, где буква b встречается только в виде массива b^n, где n — четное число.

Сначала зададим массив b^n. Это $(bb)^*$, кстати, сюда входит и пустое слово (случай $n = 0$). Слова языка — всевозможные последовательности букв a, c и таких массивов.

Искомое регулярное выражение — $(a \vee (bb)^* \vee c)^*$.

Но не все формальные языки являются регулярными.

Пример нерегулярного языка.

Рассмотрим алфавит $A = \{a\}$. Тогда язык, состоящий из слов, длина которых — квадрат некоторого натурального числа, будет нерегулярным. $L = \{a^k | k = n^2, n \in N\}$

3 Источники и языки.

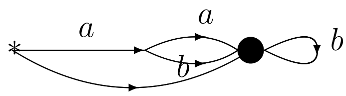
Пусть зафиксирован некоторый алфавит A. Возьмем ориентированный псевдограф, некоторым ребрам которого приписаны буквы из алфавита A. Ребра без букв назовем пустыми. Выделим некоторое множество вершин, называемых начальными и множество вершин, называемых заключительными. Такая конструкция называется *источником.*

Начальные вершины обозначаются $*$, а заключительные — $\bullet$.

Рассмотрим путь $e_1, ..., e_k$ в источнике. Выпишем последовательно буквы, приписанные ребрам $e_1, ..., e_k$. Получившееся слово назовем *словом, порожденным данным путем.* Если все ребра пути пустые, то такой путь порождает пустое слово.

Каждому источнику ставится в соответствие язык $L \subset A^*$ следующим образом. Для каждого пути из некоторой начальной вершины в некоторую заключительную выписывается порожденное им слово. Все такие слова, и только они составляют язык L. Говорят, что источник *порождает* язык L.

Пример.



$$L = a^2b^* \vee abb^* \vee b^*$$

Чтобы проверить, что данный источник порождает именно этот язык, нужно рассмотреть все пути, ведущие из начальной вершины в заключительную.

Первая теорема Клини.

Каждый язык, порождаемый источником, является регулярным.

Для доказательства будет полезна лемма об источниках.

Пусть вершины источника пронумерованы. И пусть R_{ij}^k обозначает множество всех слов, порожденных путями в данном источнике из вершины с номером i в вершину с номером j, не проходящими вершину с номером больше k. Следующая лемма иллюстрирует "понижение степени" R_{ij}^k и позволяет таким образом перейти к простейшим языкам R_{ij}^0.

Лемма.

Выполнены следующие равенства.

1)$R_{ij}^k = R_{ij}^{k-1} \vee R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

2)$R_{kj}^k = (R_{kk}^{k-1})^* R_{kj}^{k-1}$

3)$R_{ik}^k = R_{ik}^{k-1}(R_{kk}^{k-1})^*$

4)$R_{kk}^k = (R_{kk}^{k-1})^*$

Доказательство леммы.

Докажем первое равенство. Для этого рассмотрим множество путей, ведущих из вершины i в вершину j и не проходящих вершину с номером большим, чем k (но саму вершину k проходить можно). Таким образом, все пути, слова на которых составляют R_{ij}^k, могут либо вовсе не проходить вершину k, либо проходить ее некоторое количество раз. В первом случае слова на всех таких путях по определению составляют язык R_{ij}^{k-1}. Во втором случае каждый такой путь можно поделить на части: путь до вершины k, несколько путей, выходящих из k и возвращающихся в нее, и путь из k в j, причем во всех этих путях вершина k может встречаться только в начале и в конце. Таким образом, множество слов, порожденное этими путями, можно описать выражением $R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$. Поэтому $R_{ij}^k = R_{ij}^{k-1} \vee R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$. Первое равенство доказано.

Пункты 2) $-$ 4) следуют из 1). Выведем пункт 2).

Пусть $i = k$. Тогда $R_{kj}^k = R_{kj}^{k-1} \vee R_{kk}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1} = (\Lambda \vee R_{kk}^{k-1}(R_{kk}^{k-1})^*)R_{kj}^{k-1} = (R_{kk}^{k-1})^*)R_{kj}^{k-1}$. Остальные пункты выводятся аналогично.

Доказательство теоремы.

Составим регулярное выражение для языка, порожденного произвольным источником.

Итак, рассмотрим источник **И** с n вершинами. Некоторым образом перенумеруем его вершины. Множество начальных вершин обозначим I, а множество заключительных — F.

Очевидно, что вырабатываемое им множество слов есть $\vee_{i\in I,k\in F}R^n_{lk}$, это слова, порожденные путями от начальных вершин к заключительным, которые не проходят вершины с номерами, большими, чем количество вершин. Далее, для каждого R^n_{lk} применяем лемму до тех пор, пока в ней не будут участвовать лишь R^0_{ij}, то есть слова, соответствующие множествам путей из вершины i в вершину j, не заходящих ни в какую другую вершину. Можно выписать конкретные выражения для каждого R^0_{ij} следующим образом.

$R^0_{ij}=\emptyset$, если нет ребер, ведущих из вершины i в вершину j, причем $i\neq j$.

$R^0_{ij}=a_1\vee...\vee a_k$, если из i в j ведут ребра с буквами $a_1,...,a_k$. Если от i к j ведет еще и пустое ребро, то в объединение добавляется пустое слово Λ.

$R^0_{ij}=\Lambda$, если есть только пустое ребро.

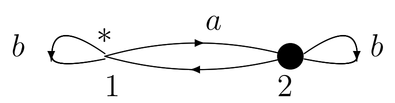
Множество R^0_{ii} также всегда содержит пустое слово Λ.

Ясно, что языки R^0_{ij} регулярны. Видно, что все языки R^k_{ij} получаются из них с помощью операций объединения, конкатенации и итерации. Следовательно, все языки R^k_{ij} регулярны, поэтому регулярен и язык $\vee_{i\in I,k\in F}R^n_{lk}$.

Теорема доказана.

Применяя лемму, можно также формализовать процесс определения языка, вырабатываемого источником.

Пример.



Этот источника порождает язык $L=R^2_{12}=R^1_{12}(R^1_{22})^*=(R^0_{11})^*R^0_{12}(R^0_{22}\cup R^0_{21}(R^0_{11})^*R^0_{12})^*=b^*a(b\vee b^*a)^*$.

Источник называется *двухполюсником*, если в нем ровно одна начальная вершина и ровно одна заключительная, причем они не совпадают, ни одно ребро не входит в начальную вершину, а из заключительной вершины не выходит ни одно ребро.

Утверждение. *Для любого источника существует эквивалентный* *ему двухполюсник.*

Доказательство.

В данном источнике все начальные и заключительные вершины сделаем обыкновенными и введем дополнительные вершины q_0 и q_f. Из q_0 проведем пустые ребра в бывшие начальные, а из бывших заключительных проведем пустые ребра в q_f. Получившийся источник — двухполюсник, эквивалентный данному.

Вторая теорема Клини.

Любой регулярный язык порождается некоторым источником.

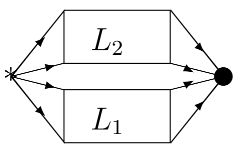
Доказательство.

Построим для регулярного языка источник.

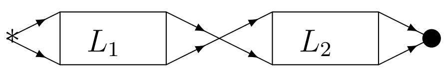
Для простейших $\emptyset, \Lambda, a$ соответствующие источники выглядят так:

Далее, пусть построены источники для регулярных языков L_1 и L_2. Можно считать их двухполюсниками.

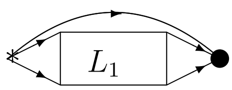
Тогда для языков $L_1 \vee L_2$ и $L_1 L_2$ источники будут:



и



А для языка L_1^* —



Так можно построить источник для любого регулярного языка.

Теорема доказана.

4 Грамматики.

Грамматика — это набор правил образования слов определенного алфавита. Здесь будут рассматриваться грамматики типа 3, также называемые автоматными грамматиками. Далее они будут называться *грамматиками.* Дадим строгое определение.

Задан алфавит $T = \{a, b, ...\}$. Его будем называть *основным* или *терминальным.*

Также введем *вспомогательный (нетерминальный)* алфавит $N = \{A, B, ...\}$.

Выделим во вспомогательном алфавите начальный символ $I \in N$ (*аксиому*). Далее зададим конечное множество выражений (*правил вывода*) следующих типов:

$A \to aB, a \in T, A, B \in N$;

$A \to a; A \to B; I \to \Lambda$.

Таким образом задана *грамматика* G.

Иногда правила для одного вспомогательного символа записываются так: $A \to \alpha_1|\alpha_2|...|\alpha_k$, что означает $A \to \alpha_1, A \to \alpha_2, ..., A \to \alpha_k$.

Пусть α — слово из алфавита $T \cup N$, имеющее вид $\alpha = \alpha_1 A \alpha_2$ и пусть в грамматике есть правило $A \to \beta$ (β может иметь вид B, aB, a или Λ). Тогда производим замену A на β и получаем $\alpha' = \alpha_1 \beta \alpha_2$.

Говорят, что α' *непосредственно выводится* из α. Обозначается $\alpha \Rightarrow \alpha'$.

Слово α' *выводится* из α, если можно указать цепочку слов вида $\alpha, \alpha_1, \alpha_2, ...\alpha_r = \alpha'$, такую, что каждое слово в цепочке непосредственно выводится из предыдущего. Обозначается $\alpha \Rightarrow^* \alpha'$

Множество выводимых из аксиомы I слов, содержащих только буквы основного алфавита T, называется *языком, порожденным грамматикой.*

Пример.

$T = \{a\}, N = \{I, B\}$

$I \to aB, I \to a, B \to aI$.

В данной грамматике слово aaa выводится следующим образом:

$I \Rightarrow aB \Rightarrow aaI \Rightarrow aaa$

Здесь последовательно применяются правила 1, 3 и 2.

Выпишем цепочки выводимых слов:

$1) I, a$

$2) I, aB, a^2I, a^3$

...

$k) I, aB, a^2I, a^3B, ..., a^{2k-2}I, a^{2k-1}$

Таким образом язык, порожденный данной грамматикой, есть множество $L = \{a^l\}$, где l — нечетное.

Утверждение.

Задаваемый грамматикой язык порождается источником, и наоборот.

Доказательство.

1. Переход от грамматики к источнику.

Пусть дана грамматика G:

$T = \{a, b, ...\}, N = \{A, B, ...\}, I \in N$ и некоторый набор правил вида $A \to aB, A \to a, I \to b$.

Построим источник, порождающий тот же язык, что и грамматика.

Каждой вспомогательной букве ставится в соответствие вершина источника, причем I соответствует начальной вершине q_0. Отдельно вводится заключительная вершина q_f.

Каждому правилу вида $A \to aB$ ставится в соответствие ребро с буквой a из вершины A в вершину B, а правилу $A \to a$ — ребро с a из A в заключительную вершину q_f. Если правило имеет вид $A \to B$ или $A \to \Lambda$, то ребра соответственно в B и в q_f проводятся пустые.

2. Переход от источника к грамматике.

Пусть L — язык, порожденный источником **И**. Построим грамматику, порождающую L.

Начальной вершине поставим в соответствие аксиому I, остальным вершинам $v_1, v_2, ...$ — различные буквы $A_1, A_2, ...$, которые будут составлять нетерминальный алфавит.

Каждому ребру вида $\overset{v_i}{\cdot} \xrightarrow{a} \overset{v_j}{\cdot}$ ставится в соответствие правило вывода $A_i \to aA_j$ ($A_i \to A_j$, если это пустое ребро). Если v_j — заключительная вершина, то добавляем еще и правило $A_i \to a$ ($A_i \to \Lambda$, если ребро пустое). Если начальная вершина является заключительной, то добавляем правило $I \to \Lambda$.

Утверждение доказано.

Упрощение грамматик.

Рассмотрим грамматику с алфавитами $T = \{a, b\}, N = \{I, A, B\}$, аксиомой I и следующими правилами вывода:

$I \to a, I \to aA, I \to bI,$

$A \to aA, A \to bA,$

$B \to a.$

Видно, что в данной грамматике нетерминальный символ B никогда не появится в слове, выведенном из аксиомы, а из символа A нельзя вывести слово только из терминальных символов. Значит, эти символы и содержащие их правила не влияют на язык, порожденный грамматикой. Поэтому данная грамматика порождает тот же язык, что и грамматика со вспомогательным алфавитом $N = \{I\}$ и двумя правилами вывода: $I \to a, I \to bI$.

Таким образом, можно упростить грамматику путем *удаления недостижимых и бесполезных символов*. Формализуем вышесказанное.

Символ $B \in N$ назовем *недостижимым*, если не существует такого выводимого из аксиомы I слова β, что $\beta = \alpha_1 B \alpha_2$, где $\alpha_1, \alpha_2 \in (T \cup N)^*$.

Символ $A \in N$ назовем *бесполезным*, если не существует такого слова $\alpha \in T^*$ из основного алфавита, что α выводится из A.

Рассмотрим аксиому I и правила вида $I \to C$ и $I \to aC$. Символы, встретившиеся в правой части таких правил, отнесем к *достижимым* символам. Включим в это множество и аксиому. Проделаем то же самое для достижимых символов. Будем рассматривать правила с достижимыми символами в левой части и добавлять символы из правой части ко множеству достижимых символов. На каком-то шаге множество достижимых символов перестанет увеличиваться. Символы вспомогательного алфавита, не вошедшие в это множество — *недостижимые*. Удалим их из грамматики, а также все правила, в которых они встречаются.

Если изобразить источник для грамматики, то недостижимые символы грамматики будут соответствовать вершинам, недостижимым из начальной вершины.

Теперь рассмотрим символы вспомогательного алфавита $A \in N$, для которых в грамматике есть правила вида $A \to a, a \in T$ или $A \to \Lambda$. Отнесем их ко множеству $\mathbf{D}$ символов, не являющихся бесполезными.

Рассмотрим правила $C \to A, C \to aA$, где $A \in \mathbf{D}, a \in T$. Добавим символ C ко множеству $\mathbf{D}$. Будем проделывать этот шаг до тех пор, пока множество $\mathbf{D}$ не перестанет увеличиваться. Символы вспомогательного алфавита, не вошедшие в $\mathbf{D}$, — бесполезные. Удалим их из грамматики вместе с содержащими их правилами.

В источнике бесполезным символам грамматики соответствуют вершины, из которых недостижимы никакие заключительные вершины.

Пример.

Рассмотрим грамматику с основным алфавитом $T = \{a, b\}$, вспомогательным алфавитом $N = \{I, A, B, C, E\}$, аксиомой I и следующими правилами вывода:

$I \to aI|bA$,
$A \to aI|aB|b$,
$B \to aB|bC$,
$C \to bC$,
$E \to a$.

Упростим эту грамматику. Символы A, B, C достижимы, так как из аксиомы выводимы слова $bA, baB, babC$. Других достижимых нетерминальных символов нет, так как ни одно правило, начинающееся с I, A, B или C, не содержит других символов из N. Оставшийся символ E — недостижимый, удалим его из грамматики. Теперь найдем бесполезные символы. Образуем множество $\mathbf{D}$ из A, так как есть правило $A \to b$, выводящее терминальное слово. Найдем правила, в которых элемент $\mathbf{D}$ встречается в правой части. Это правило $I \to bA$, поэтому $\mathbf{D} = \{I, A\}$. Дальше это множество не расширяется. Оставшиеся нетерминальные символы B, C являются бесполезными.

В итоге в упрощенной грамматике есть лишь два символа вспомогательного алфавита (I и A) и следующие правила:

$I \to aI|bA$,

$A \to aI|b$.

5 Детерминированные источники.

Рассмотрим источник со следующими свойствами:

1) ровно одна начальная вершина;

2) из каждой вершины выходит ровно $|A|$ ребер, и всем этим ребрам приписаны разные буквы алфавита A.

Такой источник называется *детерминированным.*

Для любого источника можно построить детерминированный источник, порождающий тот же язык (*эквивалентный* детерминированный источник).

Для этого применяется процесс *детерминизации* источника.

Возьмем произвольный источник **И** и построим эквивалентный ему детерминированный источник **Д** следующим способом.

Перенумеруем вершины **И**; таким образом, множество вершин можно отождествить с $\{1, 2, ..., n\}$.

Вершинам источника **Д** будут поставлены в соответствие некоторые подмножества (будем называть их массивами) множества вершин исходного источника **И**. Построим эти массивы и исходящие из них ребра с буквами алфавита A.

В качестве начальной вершины источника **Д** возьмем массив, состоящий из начальных вершин источника **И** и вершин, которые достижимы из этих начальных по путям, составленных из пустых ребер (эти пути порождают пустое слово Λ); обозначим этот массив через 1^0 и поставим ему в соответствие точку 1^0 на плоскости.

Далее, для любой буквы $a \in A$ рассмотрим всевозможные пути, выходящие из некоторой вершины массива 1^0 и порождающие слово a (если такие существуют). Множество концов таких путей обозначим через 2^0

и поставим ему в соответствие точку 2^0 на плоскости. Из вершины 1^0 проведем ребро с буквой a в вершину 2^0. Если же путей с указанным свойством не существует, то введем вершину (и точку на плоскости) f^0, соответствующую пустому множеству вершин источника **И**, и направим ребро с буквой a из 1^0 в f^0.

Это построение проведем для каждой буквы алфавита A.

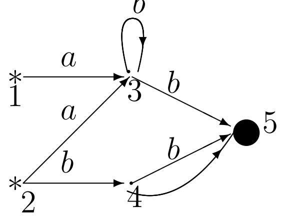
И так далее. Из вершины k^0 для каждой $a \in A$ проводится ребро a в вершину l^0, образованную номерами вершин, достижимых в **И** из вершин множества k^0 по путям, порождающим слово a. Если таких вершин нет, то из k^0 проводим ребро с буквой a к упомянутой вершине f^0. Если k^0 — пустое множество, то проводятся петли с каждой буквой входного алфавита.

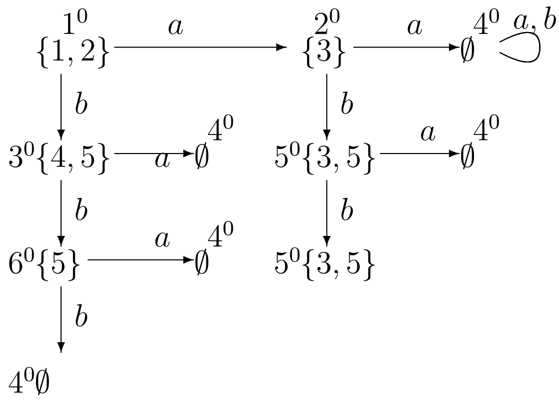
Заключительными вершинами объявляются массивы $\{i_1, ..., i_t\}$, содержащие хотя бы одну заключительную вершину источника **И**.

Из построения видно, что источник детерминированный. Также он порождает тот же язык, что и **И**, так как любому слову, приписанному пути в **И** из некоторой начальной вершины в некоторую заключительную, соответствует путь в **Д** и наоборот.

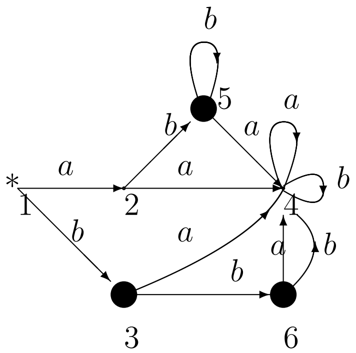
Пример.

Слева — источник **И** , справа — процесс его детерминизации.





Детерминированный источник выглядит так:



Детерминированный источник без выделенных заключительных вершин и такой, что каждому ребру сопоставлена еще и буква некоторого "выходного" алфавита, задает по определению инициальный автомат.

6 Автоматы.

Определение и способы задания.

Рассмотрим алфавит A — входной алфавит, его элементы — входные буквы, алфавит V — выходной алфавит, его элементы — выходные буквы. Составленные из этих букв слова называются соответственно входными и выходными. Также вводится Q — множество (алфавит) состояний. В дальнейшем буквы из A, V, Q будем обозначать, соответственно, через $a, b, ...; x, y, ..$ и $q_0, q_1, q_2....$

Чтобы определить автомат, нужно задать, в каких состояниях автомат будет находиться в зависимости от поступившей последовательности входных букв, и какие выходные буквы будет выдавать.

Конечным неинициальным автоматом называется пятерка

$$\mathfrak{A} = (A, Q, V, \varphi, f)\,,$$

где $Q \times A \to Q$ — функция переходов, $f : Q \times A \to V$ — функция выходов.

Если выделено начальное состояние q_0, автомат $(A, Q, V, \varphi, f, q_0)$ называется *инициальным.*

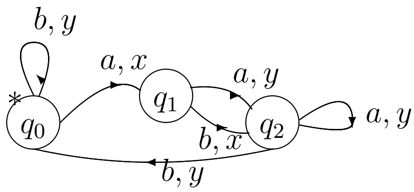
Если в момент времени t на вход автомату $\mathfrak{A}$ поступает буква $x(t)$, на выходе получается некоторая буква $y(t)$, и q_t означает состояние автомата в момент времени t, то функционирование автомата может быть задано следующей системой уравнений:

$$\begin{cases} y(t) = f(q_{t-1}, x(t)), \\ \quad q_t = \varphi(q_{t-1}, x(t)). \end{cases}$$

Также распространены способы задания в виде диаграммы Мура или таблицы.

Диаграмма Мура изображается как ориентированный псевдограф, где вершины — состояния, из каждого состояния выходят ровно $|A|$ ребер, и на каждом из них написана одна из входных букв (разным ребрам — разные буквы). Через запятую пишется выходная буква (выход при данных входной букве и состоянии, из которого выходит ребро). То есть диаграмма Мура выглядит как детерминированный источник без заключительных вершин, но с выходными значениями.

Начальное состояние (для инициального автомата) обозначается звездой.



В таблице отображаются новое состояние и выходная буква при заданных входной букве и текущем состоянии.

$x \diagdown q$	q_1	q_j	q_s
a_1			
a_i		$\varphi(q_j, a_i), f(q_j, a_i)$	
a_n			

Пример различных представлений автомата.

Рассмотрим автомат $\mathfrak{A}$, для которого все три алфавита A, V, Q совпадают с $\{0, 1\}$. Таким образом,

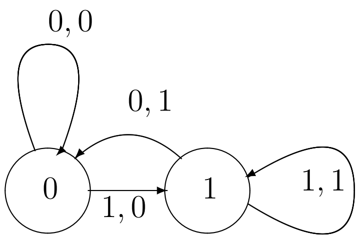
$$\mathfrak{A} = (\{0, 1\}, \{0, 1\}, \{0, 1\}, \varphi, f)\,.$$

Функции f, φ зададим формулами $f(q, a) = q$, $\varphi(q, a) = q \oplus a$, где $\oplus$ есть сложение по модулю 2.

Система уравнений для данного автомата такова:

$$\left\{ \begin{aligned} y(t) &= f(q_{t-1}, x(t)) = q_{t-1}, \\ q_t &= \varphi(q_{t-1}, x(t)) = q_{t-1} \oplus x(t). \end{aligned} \right.$$

Изобразим тот же автомат в виде диаграммы и в виде таблицы.



$x \backslash q$	0	1
0	0, 0	1, 1
1	1, 0	0, 1

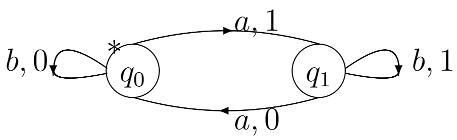
Этот автомат осуществляет задержку входной буквы на один такт.

Отображение языков, задаваемое автоматом.

Пусть дан инициальный автомат. Возьмем произвольное слово α из A^*, $\alpha = a(1)a(2)...a(r)$. Рассмотрим в диаграмме Мура путь, начинающийся в начальной вершине q_0, дальше ведущий по ребру $a(1)$, потом по $a(2)$ и так далее до $a(r)$. Выпишем последовательно буквы выходного алфавита, написанные на этих ребрах: $v(1)...v(r) \in V^*$.

Получится отображение $\Phi : A^* \to V^*$, $\Phi(a(1)...a(r)) = v(1)...v(r)$, в частности, $\Phi(\Lambda) = \Lambda$.

Пример.



$$\Phi(a^3b^2) = 10111$$

$$\Phi(a^n) = \underbrace{1010...}_{n}$$

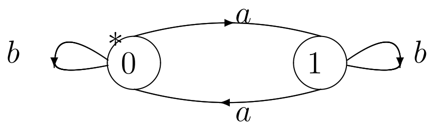
Два автомата с одними и теми же входными и выходными алфавитами называются *эквивалентными*, если задаваемые ими отображения совпадают.

Автомат Мура.

Для распознавания языков удобно использовать автомат следующего вида: для каждого состояния $q_i \in Q$ на всех ребрах, входящих в q_i, написана одна и та же буква выходного алфавита V (V-метка).

Диаграмму в этом случае можно изображать проще: состояния изображаются в виде круга, внуть которого вписана V-метка. На ребрах выходные метки не отображаются.

Например, для автомата из предыдущего примера муровский автомат выглядит так:



Теорема.

Для любого автомата можно построить эквивалентный ему муровский автомат.

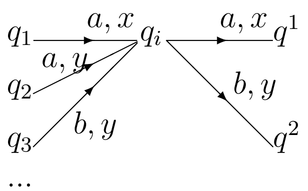
Доказательство.

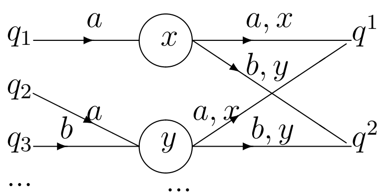
Рассмотрим состояние q_i и ребра, в него входящие. Возможны два случая:

1) На всех входящих ребрах V-метки одинаковые. Назовем такое состояние простым. Тогда стираем метки на ребрах и пишем это выходное значение в состоянии. Таким образом, простому состоянию обычного автомата соответствует одно состояние муровского автомата.

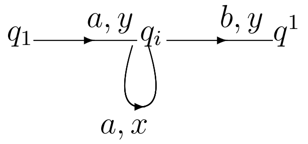
2) V-метки разные. Назовем такое состояние сложным. Пусть число различных V-меток, написанных на ребрах, входящих в вершину q_i, равно k. Тогда в муровском автомате состоянию q_i будет соответствовать ровно k состояний. В каждом из k состояний своя V-метка, и в него ведут ребра, входившие в q_i в исходном автомате с этой V-меткой. q_i стираем, а ребра, выходившие из него, выводим из каждой новой вершины.

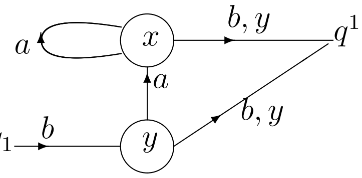
Так преобразуется каждое сложное состояние q_i:





В частности, если в сложном состоянии есть петля, расщепление происходит так:





Процедура проделывается для каждого состояния. В итоге получается муровский автомат, эквивалентный данному.

Язык, создаваемый муровским автоматом.

Рассмотрим инициальный автомат с выходным алфавитом $B = \{0, 1\}$ и произвольным входным алфавитом A. Построим для него детерминированный источник, взяв тот же ориентированный псевдограф. Состояния муровского автомата, в которых написано 1, назовем *заключительными* вершинами. Начальная вершина — q_0 (начальное состояние автомата). Этот источник порождает некоторый язык L.

Итак, любой автомат с выходным алфавитом $\{0, 1\}$ задает детерминированный источник, и ясно, что это соответствие взаимно-однозначно. Таким образом, указанный автомат задает язык $L \subseteq A^*$. (Говорят также, что автомат *распознает* или *допускает* язык.) Язык, для которого существует распознающий его автомат, называется *автоматным.*

Таким образом, автомат распознвет язык L с помощью выделенного символа выходного алфавита: слова из L он преобразует в выходные слова, которые заканчиваются на 1, а слова не из L — в слова, заканчивающиеся на 0.

Язык, распознаваемый рассмотренным выше муровским автоматом — все слова, в которое a входит нечетное число раз.

Переформулируем теоремы Клини следующим образом.

Теорема Клини (для автоматов)

1) Любой регулярный язык является автоматным.

2) Любой автоматный язык является регулярным.

Таким образом, совпадают следующие классы языков:

1) регулярные,

2) автоматные,

3) задаваемые грамматикой.

Причем в каждом случае представление языка не единственно. Но для конечных автоматов существует алгоритм, находящий оптимальный по числу состояний автомат. Этот алгоритм будет описан в следующем разделе.

7 Минимизация автоматов.

Так как существуют различные автоматы, задающие одинаковые отображения Φ, возникает вопрос, как построить автомат, эквивалентный данному, но с меньшим числом состояний. Процесс минимизации позволяет получить автомат с наименьшим числом состояний.

Состояния q_i и q_j некоторого автомата называются *эквивалентными*, если инициальные автоматы с начальными состояниями q_i и q_j эквивалентны.

Разобьем множество состояний автомата Q на классы эквивалентности (эквивалентные состояния находятся в одном классе, состояния из разных классов неэквивалентны). Каждый класс объявляется состоянием нового автомата $\widetilde{q}_1, \widetilde{q}_2, ..., \widetilde{q}_s$. От состояния $\widetilde{q}_i$ к состоянию $\widetilde{q}_j$ проводится ребро с буквами a, v, если такие ребра были между состояниями исходного автомата из соответствующих классов.

$\widetilde{q}$ — начальная вершина, если в соответствующем классе содержится начальная вершина.

Полученный автомат называется *минимальным* или *приведенным.*

Разбиение на классы происходит следующим образом.

Алгоритм минимизации.

1) q_i, q_j отнесем к одному классу, если $f(q_i, a) = f(q_j, a)$ для каждой буквы входного алфавита. Полученный класс назовем классом 1-эквивалентности.

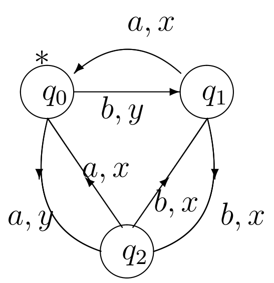
2) Каждый полученный класс разобьем следующим образом:

q_s, q_t из класса 1-эквивалентности находятся в одном классе 2-эквивалентности, если для каждой буквы входного алфавита a состояния $\varphi(q_s, a)$ и $\varphi(q_t, a)$ находятся в одном классе 1-эквивалентности для каждого a.

i) q_s, q_t из одного класса $(i-1)$-эквивалентности находятся в одном классе i-эквивалентности, если для каждого a состояния $\varphi(q_s, a)$ и $\varphi(q_t, a)$ находятся в одном классе $(i-1)$-эквивалентности.

Когда этот процесс остановится (классы прекращают делиться), получится требуемое разбиение. Легко видеть, что максимальное число шагов данного алгоритма на единицу меньше числа состояний исходного автомата.

Пример.

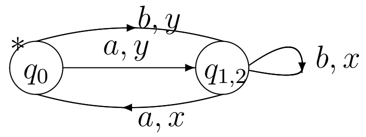


	q_0	q_1	q_2
a	q_2, y	q_0, x	q_0, x
b	q_1, y	q_2, x	q_1, x

На первом шаге алгоритма (проверка совпадения выходных символов) выделяются два класса: $I = \{q_0\}, II = \{q_1, q_2\}$. Рассмотрим $\varphi(q_i, a_j)$ для второго класса.

	q_1	q_2
a	I	I
b	II	II

Функции переходов переводят состояния в одинаковые классы, дальше разбиения не происходит. Значит, состояния q_1 и q_2 эквивалентны. Минимальный автомат имеет вид:

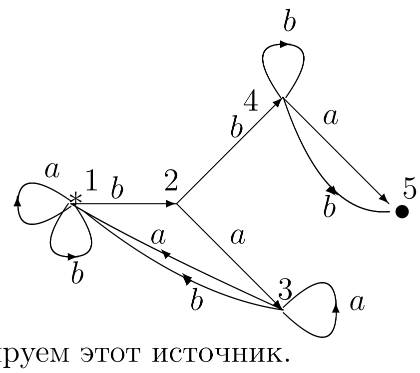


Пример.

Задание: по грамматике построить источник, детерминировать его, получить автомат, распознающий тот же язык, минимизировать полученный автомат и записать грамматику для минимального автомата.

$I \to aI|bI|bA,$
$A \to aB|bC,$
$B \to aB|bI|aI,$
$C \to bC|a|b.$

Источник, порождающий тот же язык, что и грамматика.



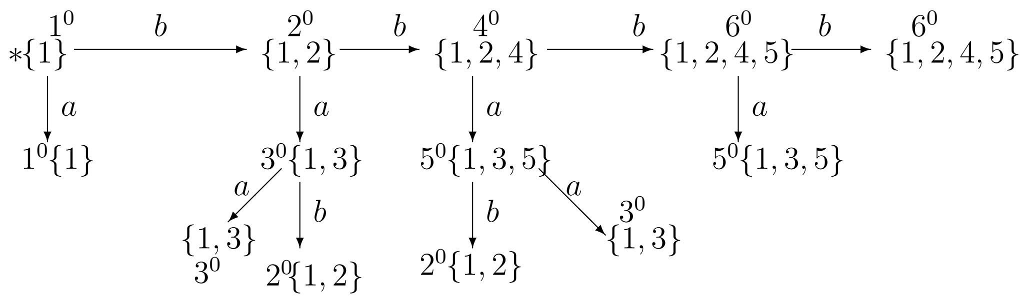
$1 = I$

$2 = A$

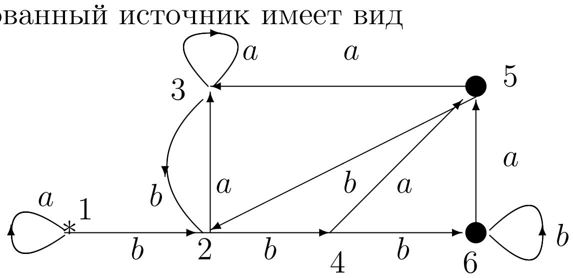
$3 = B$

$4 = C$

Детерминируем этот источник.



Детерминированный источник имеет вид



Такому источнику можно поставить в соответствие муровский автомат, допускающий тот же язык с помощью выходного символа 1.

	1	2	3	4	5	6
a	1,0	3,0	3,0	5,1	3,0	5,1
b	2,0	4,0	2,0	6,1	2,0	6,1

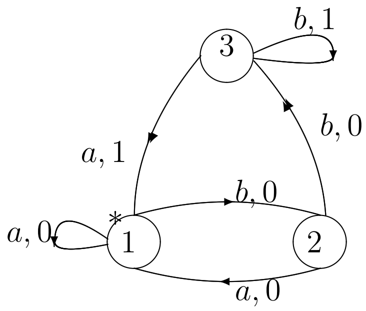
На первом шаге выделяютя классы 1-эквивалентности:
$I = \{1,2,3,5\}, II = \{4,6\}$

	1	2	3	4	5	6
a	I	I	I	I	I	I
b	I	II	I	II	I	II

Классы 2-эквивалентности:
$I = \{1,3,5\}, II = \{2\}, III = \{4,6\}$

	1	2	3	4	5	6
a	I	I	I	I	I	I
b	II	III	II	III	II	III

Минимизация закончена. Получившийся автомат таков:



Порождающая грамматика имеет вид
$I \to aI|bA, A \to aI|bB, B \to bB|aI|a|b.$

8 Типовой расчет.

Задача №1.

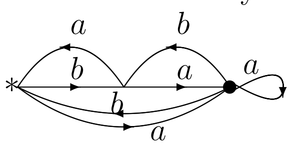
По словесному описанию языка составить регулярное выражение. Составить грамматику, порождающую данный язык и выписать вывод какого-нибудь четырехбуквенного или пятибуквенного слова.

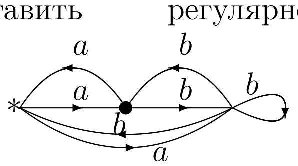
1) Каждое слово языка содержит подслово abb или aac.
2) Все слова языка имеют длину не менее 3 и не начинаются с буквы c.
3) Длина каждого слова не меньше 2, и вторая буква всегда b.
4) Перед каждой буквой c стоит a.
5) Буква c может встречаться только как часть подслова acb.
6) В каждом слове содержится не более двух букв c.
7) Ни в одном слове буква b не следующая после c.
8) Ни в одном слове нет двух и более букв a подряд.
9) В словах языка нет ни подслова ba, ни подслова bb.
10) После буквы a в слове языка всегда идет bc.
11) Длина каждого слова не меньше 2, предпоследняя буква каждого слова - c.
12) Ни одно слово не содержит подслова ab.
13) Буква c не встречается ни в одном слове раньше («левее»), чем

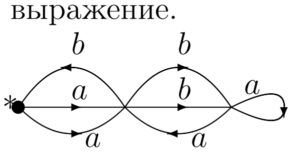
встретится хотя бы одна буква a.
14) На всех нечетных местах каждого слова находится a.
15) Если слово начинается на ab, оно заканчивается на c.
16) Если в слове есть буква a, то она написана не менее 2-х раз подряд.
17) На четных местах каждого слова находится b, пустого слова нет.
18) В словах языка после буквы a всегда идет bb.
19) Перед каждой буквой c веда стоит aa.
20) Ни одно слово не содержит подслова cc.
21) Если в слове есть буква b, то есть и a.
22) Язык состоит из всех слов четной длины.
23) В каждом слове 3 буквы c, одна из них в конце.
24) Язык состоит из всех слов нечетной длины.
25) Каждое слово содержит не менее 3-х букв a.
26) Ни одно слово не содержит подслова cb.
27) Буква b не встречается «правее» буквы a ни в одном слове.
28) На четных местах каждого слова находится буква c.
29) В каждом слове — не менее 4-х букв b.
30) Все слова начинаются и заканчиваются на одну и ту же букву.

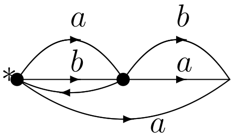
Задание №2

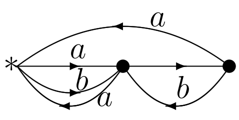
По источнику составить регулярное выражение.

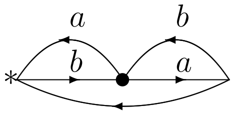
1) 

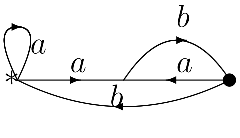
2) 

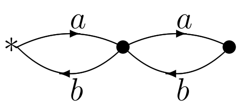
3) 

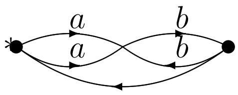
4) 

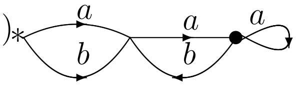
5) 

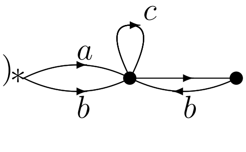
6) 

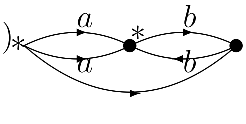
7) 

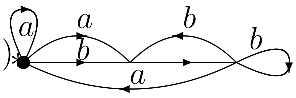
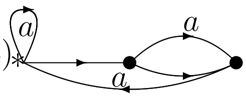
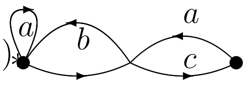
8) 

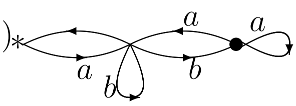
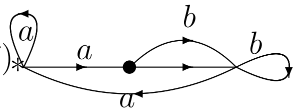
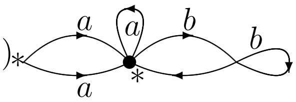
9) 

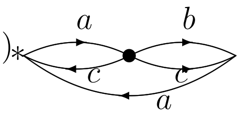
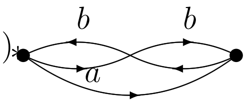
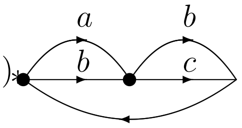
10) 

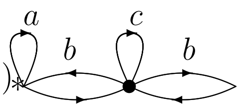
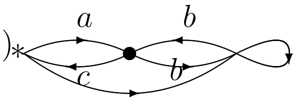
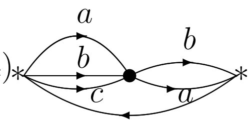
11) 

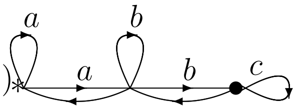
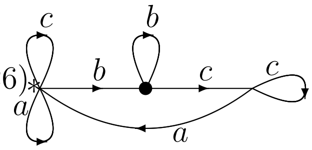
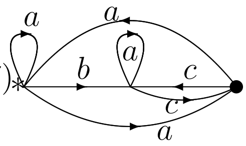
12) 

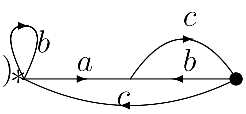
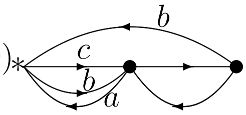
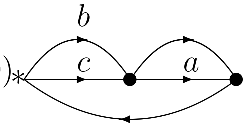
13)  14)  15) 

16)  17)  18) 

19)  20)  21) 

22)  23)  24) 

25)  26)  27) 

28)  29)  30) 

Задание №3

1) По данной грамматике составить источник;
2) детерминировать его;
3) получить автомат, представляющий тот же язык, минимизировать полученный автомат;
4) записать грамматику для минимального автомата.

1) $I \to a|aA|aB, A \to aA|b, B \to aC|aA|bC, C \to bC|b;$

2) $I \to aA|aB|bI, A \to bB|\Lambda, B \to aI|aC|a, C \to a|bB|b;$

3) $I \to aA|aB|aC, A \to aB|bA|bC, B \to a|b, C \to aC|bB|\Lambda;$

4) $I \to aA|b|\Lambda, A \to aB|aC|bI, B \to a|b, C \to a|bB|b;$

5) $I \to aA|aB|bB, A \to aA|bI|\Lambda, B \to aC|bB, C \to bB;$

6) $I \to aA|aB|a, A \to a|bA, B \to aC|bA|bC, C \to bC|b;$

7) $I \to aI|bA|bB, A \to aB|b, B \to bI|bC|b, C \to a|bB|b;$

8) $I \to aA|aB|aC, A \to aA|aC|bB, B \to a|b, C \to aB|bB|bC;$

9) $I \to aB|bA|bB, A \to aA|a|bI, B \to aB|bC, C \to aB|A;$

10) $I \to b|bA|bB, A \to a|bA, B \to aC|bA|bC, C \to aC|a$

11) $I \to aI|bA|bB, A \to aB|\Lambda, B \to bI|bC|b, C \to a|aB|b;$

12) $I \to bA|bB|bC, A \to aA|aC|bB, B \to a|b, C \to aB|bC|\Lambda;$

13) $I \to a|bA|\Lambda, A \to aI|bB|bC, B \to a|bC, C \to a|aB|b;$

14) $I \to aB|bB|bA, A \to aI|bA|\Lambda, B \to aB|bC, C \to aB;$

15) $I \to bA|bB|b, A \to aA|b, B \to aA|aC|bC, C \to aC|a;$

16) $I \to aA|aB|bI, A \to a|bB, B \to aI|aC|a, C \to a|aB|b;$

17) $I \to bA|bB|bC, A \to aB|bA|bC, B \to a|b, C \to aB|aC|bB;$

18) $I \to aB|aA|bB, A \to aI|bA|b, B \to aC|bB, C \to bB|A;$

19) $I \to aA|aC|a, A \to aB|aC|bB, B \to bB|b, C \to aC|b;$

20) $I \to aB|aC|bI, A \to a|bC|b, B \to bC|\Lambda, C \to a|bC|b;$

21) $I \to aA|aB|aC, A \to a|b, B \to aB|bA|\Lambda, C \to aA|bB|bC;$

22) $I \to aB|b|\Lambda, A \to a|bC|b, B \to aA|aC|bI, C \to aA|b;$

23) $I \to aA|aC|bA, A \to aB|bA, B \to bA, C \to aC|bI|\Lambda;$

24) $I \to aB|aC|a, A \to bA|b, B \to a|bB, C \to aA|bA|bB;$

25) $I \to aI|bA|bC, A \to bI|bB|b, B \to a|bA|b, C \to aA|b;$

26) $I \to aB|aC|aA, A \to aC|bA|bC, B \to aA|aB|bC, C \to a|b;$

27) $I \to aA|bA|bC, A \to aA|bB, B \to aA|C, C \to aC|a|bI;$

28) $I \to bA|bB|bC, A \to a|b, B \to bB|aA|\Lambda, C \to bA|aB|aC;$

29) $I \to bA|bC|b, A \to aB|bB|bC, B \to aB|a, C \to bC|a;$

30) $I \to bI|aA|aC, A \to aI|aB|a, B \to a|aA|b, C \to bA|a.$

9 Литература

1.*Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции: Пер. с англ. М.: Мир, 1978.
2. *Белоусов А.И., Ткачев С.Б.* Дискретная математика. М.: Изд-во МГТУ им. Н.Э.Баумана, 2001.
3. *Кудрявцев В.Б., Алешин С.В., Подколзин А.С.* Введение в теорию автоматов. М.: Наука, 1985.
4. *Трахтенброт Б.А., Бардзин Я.М.* Конечные автоматы (поведение и синтез). М.: Наука, 1970

Содержание

1 Предварительные сведения. 2

2 Регулярные языки. 2

3 Источники и языки. 4

4 Грамматики. 8

5 Детерминированные источники. 11

6 Автоматы. 13

7 Минимизация автоматов. 17

8 Типовой расчет. 21

9 Литература 25